

Information Integrity In End-user Systems

D. Chadwick, J. Knight, P. Clipsham
Information Integrity Research Group,
School of Computing & Maths Sciences, University of Greenwich,
Woolwich, London SE18 6PF,
United Kingdom
Phone: +44 0181 331 8509 Fax: +44 0181 331 8665
Email: D.R.Chadwick@gre.ac.uk

Abstract

Much research has been undertaken towards determining why programming errors occur during systems development and this has resulted in improved techniques for use by practitioners and teachers. However, little research has been done on errors occurring in end-user applications such as spreadsheets that are not developed under formal system development guidelines. The approach taken in the research reported here is that many of these errors are due to an education process that concentrates on teaching 'how to do things correctly' but ignores 'how to avoid doing things incorrectly'.

Research into the development of spreadsheets has identified common errors made by endusers in industry and by students whilst learning. It has been shown that there is a correspondence between industry and education in the types of errors made. From this has developed an approach which, in addition to teaching correct methods of building systems, implicitly addresses the common errors identified above.

Keywords

Spreadsheets, Errors, End-users, Education, Auditing

1 INTRODUCTION

The problem of data integrity in an information process pervades the whole information system including end-user data entry and retrieval. Although much attention has been devoted to verification and validation of information processes, in education and industry research has shown that there is a persistent problem in end-user errors where typically data integrity is reduced.

Over several years of teaching spreadsheets to students it has become apparent that each new intake of students tends to make the same errors as the year before. However, it has been difficult to give feedback to students as no tools existed for aiding the marking of spreadsheets - a lecturer would browse the spreadsheet looking at those areas he/she considered from personal experience might contain errors. It was realised that better methods needed to be investigated that would:

- aid the students in making fewer errors whilst building a spreadsheet
- aid the lecturer in assessing the students spreadsheet for correctness.

A search was made for methodologies and tools available in industry which might be useful in an educational environment. A pilot investigation revealed that, in industry, large scale software developments by professional computing staff were subject to formal development methods and monitored by auditors for errors throughout their life-history. However, small scale applications, such as spreadsheets developed by endusers, were subject to no such developmental methodologies and only ad-hoc monitoring by audit staff. There was anecdotal evidence (but little research evidence) that errors were not only numerous but could be in existence for some time before being detected and corrected. Under these circumstances, it seemed reasonable to assume that a cost had been borne by the company; a cost that perhaps was far in excess of the direct cost of correcting the error as business decisions may have been made using the incorrect data. This is recognised in the general auditing literature :

‘the correction of errors resulting from serious faults in the introduction of a new or modified system can itself be extremely costly, and a great deal of damage can be done before correction is achieved’ (Coates et al, 1989, p254)

And also by a standard text used for teaching Computer auditing in the UK. The book has has one paragraph of four lines devoted to spreadsheet errors:

‘Users of spreadsheets and database packages are necessarily immune from errors. While it is true that such packages provide a safe processing environment within which it is difficult if not impossible to make undetected or obscure input and output errors, it is still possible to make errors in logic. In fact, such errors maybe more difficult to detect than they would have been if a procedural programming language had been employed... This may be the case if a very

large spread sheet is generated, so that only small parts of it can be viewed on screen at any one time... (Chambers A.D, 1996, p151)

The third line admits that the problem for spreadsheets may be even worse than for normal software particularly with logical errors. Despite this, there is no literature on how to prevent errors or on how to conduct a spreadsheet audit.

2 OBJECTIVES

It was reasoned that new ideas might not only help students but could possibly aid endusers in industry. In addition, anything that improved the quality of spreadsheets would also be useful to computer auditors who might wish to audit them. One objective of the research, therefore, was to establish whether the errors of students and endusers in industry were similar so that valid comparisons could be made. If this were the case, then methods and tools designed for students may also find a use in industry.

3 RESEARCH METHOD

It was proposed to collect data from the following sources:

- teaching staff and students at the University of Greenwich
- professional computer auditors via their professional bodies
- existing literature in the form of educational books (used for teaching spreadsheets), research papers and journal articles from the professional auditing bodies.

From lecturing staff and students both quantitative and qualitative data was required. Internal emails were sent to staff in the Computing and Business schools to specifically reach those who taught spreadsheets. Lecturers generally have a 'feel' for the types of errors students make and, although not quantifiable, such anecdotal evidence is useful. For hard quantitative data, a test was given to first year computing students, consisting of multiple-choice questions in which they were asked to choose the appropriate Excel formulae for specified cells on a given spreadsheet which contained no data. This paper-based test was decided upon in preference to a live test, using an active spreadsheet on computer, in order to eliminate the 'trial and error' approach that characterises much student work (this is where they enter what they think looks like a good formula, view the result, and if the result looks reasonably good they accept that formula without further consideration).

Professional Computer Auditors were reached through the relevant professional bodies. An article was sent to the editor of the journal of the British Computer Society Computer Audit Specialist Group. The editor gave support to the project

and the article was published in the Summer 1997 edition. A similar article was sent to the editor of "Internal Auditing", the journal of the Institute of Internal Auditors of the United Kingdom (IIA-UK). This was published in the May 1997 edition.

Little literature was found which directly addressed the problem of spreadsheet errors. Searches were made of journals in the areas of Human-Computer Studies, Computer Audit, Internal Auditing and Accountancy (see Reference List).

4 IS THERE A PROBLEM AT ALL ?

The first task was to determine if the problem was recognised in the research literature.

'Spreadsheet applications are perhaps more vulnerable to poor design and to errors than conventional programs... As a result, spreadsheet applications often have problems,...One experimental study found almost half of the simple spreadsheet applications created by experienced users contained errors.' (Davis S. 1996)

In the 19th June 1997 edition of Computer Weekly magazine an article appeared discussing a report by KPMG Management Consultancy on the problem of spreadsheet errors :

'Endusers are putting their companies at risk by setting up spreadsheets without realising that this demands the discipline of traditional programming' (Kavanagh J. 1997)

Further on in the same article is stated:

'Our findings are disturbing, but they are not really surprising, as 78% of models had no formal quality assurance to ensure they were built to specified requirements and were fit for purpose'

It appeared, therefore, that of the little literature that existed on the topic, there was strong support for the notion that there was a problem with spreadsheet errors in business organisations.

5 COMMON ERRORS

It may be assumed that if an operative is well trained in a task then they would make few mistakes. There is anecdotal evidence, but no research evidence, to suggest that many end-user spreadsheet builders in companies are either self-taught or taught in-house by non-professional trainers. This poses the 'little learning is a dangerous thing' problem in that such users may consistently overrate their own

abilities and thereby make more errors. This is a common problem with new students and is supported by research conducted into comparing self-appraisal and objective tests of learners abilities (Van Vliet et al 1993); this indicates that novices of both sexes consistently overrate their own computer literacy skills and hence make more errors. But even professionally trained personnel make mistakes. An assertion made by the authors is that even when training is given it too frequently concentrates on 'how to do things correctly' and often ignores 'how to avoid doing things incorrectly'. One possible way of 'avoiding doing things incorrectly' is to make spreadsheet builders aware of the common errors they are more likely to make.

5.1 Classification of Spreadsheet Building Skills

Some work has been done on errors in human-computer interaction. Batra and Sein (Batra D.1994) describe work by Norman (1983) which proposed that the human-computer interaction could be represented by:

intention -----> action -----> goal

Norman proposed two types of errors : slips and mistakes. Slips are errors that occur when the intention to act fits the intended goal but the action is not carried out according to plan; mistakes are errors that occur when an action is carried out as intended but the action itself is not appropriate to the task.

Another useful classification of errors is to divide the skills required into two categories *Planning Skills* and *Enabling Skills*.

5.2 Enabling Skills

Enabling skills are those needed to permit the user full use of the functions and capabilities of the particular spreadsheet package in use. In many ways, these are the kinds of skills that, in conventional software development, would be called 'software engineering' skills

Generic Enabling Skills are those which give a general understanding of spreadsheet principles and concepts regardless of the particular product in use. These are analogous, in normal software development, to skills associated with understanding basic programming structures regardless of the actual programming language being used.

Examples of generic spreadsheet enabling skills would be:

- understanding of the basic concept of a matrix numbered rows and alphabetically coded columns and their interaction to form individually identifiable cells used for storing data and formulae
- understanding of the concepts of formulae that perform calculations on absolute value (numbers and text strings forming part of the formula itself) and also on

referenced values (numbers, text strings being values referenced by the individual code of the cell where they are stored).

Specific Enabling Skills are those which enable the user to use the functions of the specific database package. These are analogous, in normal software development, to skills associated with understanding the constructs, reserved words and syntax of a specific programming language.

5.3 Planning Skills

The matrix of data and formulae that constitute the recognised spreadsheet model are an electronic representation of a business function in the real world. Planning skills are those required to analyse the business function in order to design the conceptual data model which is to be represented by the electronic spreadsheet model. In many ways these skills are analogous to the types of skills that in conventional software development would be termed 'requirements analysis' and 'data modelling'.

Generic Planning Skills are those that enable the user to identify business functions which are suitable for modelling with a spreadsheet and how this modelling is to be done. Analytical skills of this type need a thorough knowledge of business functionality and requirements not just for the present but also for the future. In many ways, these skills are analogous to the types of skills that in conventional software development would be termed 'requirements analysis'.

Specific Planning Skills are those required to design the specific data model for that particular business application. These skills are analogous to the types of skills that in conventional software development would be termed 'data modelling' and HCI (Human Computer Interface) issues. With spreadsheets such skills would cover:

- the HCI issues ie. what should be the best design and layout to enable the eventual user to minimise misreadings and mis-comprehension of the data and minimise invalid data input
- the data integrity issues: what should be the correct formulae and data processing macros to give data correctness to the model ?

The research findings indicate that errors occur in all the categories above but although errors in enabling skills are by far the more numerous, it is the errors in the planning skills that are the more serious.

6 ANALYSIS OF COMMON ERRORS

The first task was to establish what kind of errors were being made by students and end-users and whether there was a correspondence between the two.

6.1 Results of Surveys

The spreadsheet test given to students was analysed question-by-question. This was the most definitive source of data as the questions had been carefully framed, there were 120 students in the sample and the test was multiple choice thereby removing any subjectivity in the marking.

Of the Professional Computer Auditors, 23 responses were received from all parts of the UK from practising computer auditors or computer audit managers.

6.2 Errors In Enabling Skills

SUM incorrect use problem

Industry respondents : 80%

Students: 30%

Lecturers : three involved in teaching spreadsheets

The Microsoft Excel Help function describes the SUM function as follows:-

SUM(number1,number2, ...)

number1, number2, ... are 1 to 30 arguments for which you want the sum

A common error is to enter any formula within the SUM brackets as though the SUM was mandatory for defining a formula e.g. SUM(A12/F12) which should really be =A12/F12. The function does not produce an error and does the calculation correctly as 1 argument is acceptable. However, this is logically wrong as the formula shown above does not have any mathematical summing within it.

Relative and Absolute Copy problems

Industry respondents : 50%

Students: 40%

Lecturers - three involved in teaching spreadsheets

This is not just a problem for novices - the industry respondents indicate that problems arise even for experienced spreadsheet builders. The relative copy causes cell references in a copied formula to alter row and column references relative to the original cell copied. Novices find this difficult to grasp at first and then become blasé about it - blasé to the point where they appear to think the software will adapt the cell references wherever they happen to copy. This problem is recognised in existing research literature:

'novices experience difficulties in learning about relative and absolute cell references, a feature of all spreadsheets' (Hendry D.G and Green T.R.G 1994) referring to a 1990 paper by Doyle)

Overwriting problem

Industry respondents : 5%

Students: 0%

Lecturers - one involved in teaching spreadsheets

One of the industry respondents, a computer auditor, found this to be the major source of errors. Further analysis of this problem seems to indicate situations where

naive users, whose only duty is data entry, are not aware of the formula structure of the spreadsheet (the underlying data model) .

Circular References

Industry respondents : 10%

Students: 10%

Lecturers - three; all said it was common

This error frequently occurs in totals e.g. in Figure 1 the formula in C9 being SUM(C5.C9) will not work as the formula uses its own value in its calculation. This error will actually give a run-time error message and so probably occurs infrequently. However, it appears that students who make this error often seem incapable of grasping what a 'circular reference' is even when explained several times.

6.3 Errors In Planning Skills

The TOTALS Problem

Industry respondents : 50%

Students: 40%

Lecturers - three involved in teaching spreadsheets

In Figure 1 the total of column E in cell E9 should be SUM (E5.E8) and should contain the same data value as SUM (C9.D9). Industry respondents who cited this error mentioned the lack of cross-checking which users could easily have done. A good test, therefore, would have been to enter in cell E9 the following formula: IF(SUM (E5.E8) <> SUM (C9.D9), "Error", SUM (E5.E8)). This would show an error message when the column total and the row total were dissimilar.

Figure 1

	A	B	C	D	E	F	G
1	Lazy Days Staff Budget Costs 1995-1996						
2		Staff	Day Work	Night Work	Total	Average	Night
3		Numbers	Wages £	Wages £	Wages £	Wage £	Wages
4							% of Total
5	Managers	1	17700.5	0	17700.5		
6	Grade 1	3	45540	1400.55	46940.55		
7	Grade 2	9	122340	2000	124340		
8	Grade 3	12	102350.25	0	102350.25		
9	Grand Totals	25	287930.75	3400.55			
10							

AVERAGE problem

Industry respondents : 30%

Students: 90%

Lecturers - three involved in teaching spreadsheets

In Figure 1, to calculate the average wage in F5, the formula should be $=E5/B5$ to calculate the average wage per person. However, 90% of students thought it should be $=AVERAGE(C5:D5)$ which calculates the average of Day Work Wages and Night Work Wages. A sample of 5 students were interviewed about this error and all insisted that as an average was required then the average function should be applied with little understanding as to whether it was appropriate or not. The auditor respondents mentioned other built-in functions that were applied incorrectly.

Percentage Problem

Industry respondents : 50%

Students: 90%

Lecturers - three involved in teaching spreadsheets

In Figure 1 in G6 a formula is needed to calculate the Night Work Wages as a percentage of Total Wages. A correct formula would be $D6/E6*100$ or $D6*100/E6$. However, this presupposes knowledge of BODMAS (Brackets, Of, Division, Multiplication, Addition, Subtraction) by which the spreadsheet identifies precedence in calculations. Many students submitted $E6/D6*100$, $E6*100/D6$ and even $E6*D6/100$. All of these solutions show ignorance of basic algebra and what a percentage actually is.

Money Format:Commas, Standard Decimalisation

Industry respondents : 10%

Students: 20%

Lecturers - one involved in teaching spreadsheets

In Figure 2 we see a common problem when the cell format is specified as General on the spreadsheet. The figures have varying decimal places and make it difficult to identify a number that is incorrect by a magnitude of 10 e.g. is the value in C8 greater than the value in C9? It cannot possibly be as C9 is the Total of the column including C8 but it may appear to be at a quick glance. A possible solution to this would be a standard notation used throughout the organisation to specify cell format, decimal places and (if appropriate) commas to denote thousands.

6.4 Correlation Between Industry and Education

The analysis of the errors does indicate that similar errors are made by endusers in industry and university students.

Figure 2

	A	B	C	D	E	F	G
1	Lazy Days Staff Budget Costs 1995-1996						
2		Staff	Day Work	Night Work	Total	Average	Night
3		Numbers	Wages £	Wages £	Wages £	Wage £	Wages %
4							of Total
5	Managers	1	17700.5	0	17700.5	8850.25	0
6	Grade 1	3	45540	1400.55	46940.55	23470.275	2.98366764
7	Grade 2	9	122340	2000	124340	62170	1.60849284
8	Grade 3	12	102350.25	0	102350.25	51175.125	0
9	Grand Totals	25	287930.5	3400.55	291331.3	145665.65	1.16724499
10							

7 EXISTING METHODOLOGIES AND TOOLS

The research initiative then proceeded to look at possible spreadsheet methodologies or tools to help prevent and detect errors.

The aim of a good spreadsheet methodology would be to help the builder ensure that the spreadsheet model was a correct interpretation of the underlying data model, to help eliminate the common errors identified above and to structure the spreadsheet model for easy checking. The authors looked at ten new books on sale and ten books in a University library and presumably in use on courses. None of the books exhibited any formal methodology whatsoever. The nearest to an approach was in *Spreadsheet Applications In Financial Accounting* by Angelo E. DiAntonio (DiAntonio) which, although some years old, did attempt a structured although not detailed approach.

7.1 Di Antonio’s Methodology

DiAntonio proposed the following steps:

1. Define the Problem

It is well to spend a few minutes to understand the program.

2. Isolate the Facts

Divide the spreadsheet matrix into two areas - one for the Facts, one for the Solution. Isolate the facts. With our technique, one area of the (spreadsheet) matrix will be devoted to the facts. You may write your facts on paper or place them directly on the spreadsheet. Write them in the Facts area of the screen.

3. Format The Solution

Another portion of the (spreadsheet) matrix will display your solution to the problem. In the Solution area of the screen you will be using the data in the Facts area.

4. Test The Program

Prepare several sets of facts with solutions you know are correct.

5. Evaluate The Program

Print the program and determine if it meets your expectations. Examine the headings, labels and the format.

6. Document The Program

Documentation may be included in a special area of your matrix or prepared in hard copy and kept in a binder. This should assist both user and designer and include the program rationale and objectives.

It is well to prepare a log for changes that are made to the program. The log should include author, date, the nature of the change and the reason for the change. The extra time that you spend to document your program will result in a big payoff as you use, evaluate, and change the template.

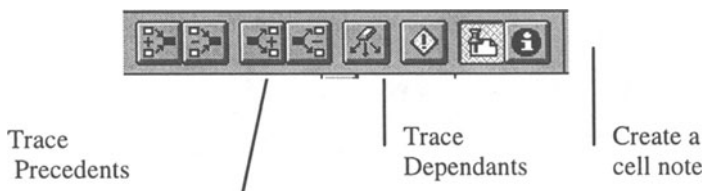
DiAntonio's approach gives spreadsheet layouts as seen in Figure 3 (the arrowed line and the Cell Note box will be explained later).

This approach gives definite steps to be followed and suggests a useful splitting of the screen into a Documentation area, a Facts area and a Solution area. The mention of testing is also beneficial as it does remind the builder that he/she needs to check for errors. However, it suggests no guidelines on testing nor any strategy for self-checking of the spreadsheet. Even so, ad-hoc testing alone may lead to detection of 'slips' but without proper direction is unlikely to detect more serious mistakes e.g. there is no suggestion in DiAntonio's approach that each of the formulae be rechecked for correctness even though formulae are a common source of errors.

7.2 Microsoft Excel Audit Tool

One factor that recurred with auditor respondents was that endusers exhibited general ignorance of the built-in auditing tools of Excel.

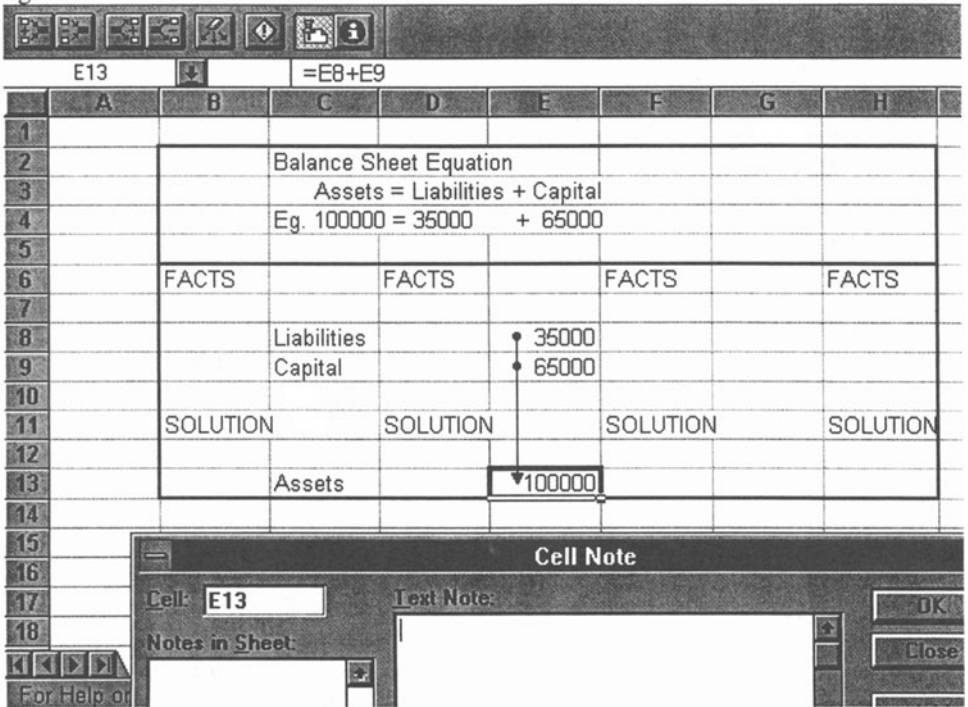
In Excel version 5.0 Microsoft (Microsoft Corporation 1994) there is included a useful audit tool found under Tools, Auditing. The Audit toolbar has the following buttons:



- Trace Precedents: for any cell formula it will show arrows pointing at the cells from which data is referenced
- Trace Descendants: for any cell it will show arrows pointing at the cells which contain formulae that reference the data in it.
- Cell Note: allows a note to be attached to a cell.

The tool can be used as shown in Figure 3. The arrowed line from E8 to E13 shows that the formula in E13 is dependent on data in E8 and E9. Also, a cell note has been opened for cell E13 for the entry of supporting text to clarify the meaning of the data in cell E13..

Figure 3



8 A PROPOSED METHODOLOGY AND TOOL

8.1 An Outline

DiAntonio's approach and the Microsoft Excel Audit tool were useful in suggesting four useful features of any workable methodology which would facilitate self-checking or auditing:

- the usefulness of a stepped approach giving a certain intellectual discipline to the process;
- the usefulness of dividing up the screen into logical parts making understanding easier for the designer, the user and the auditor;
- the usefulness of on-screen documentation (via a cell note) enabling the designer and the auditor to observe how the spreadsheet model related to the underlying data model;
- the need for graphical display to show how formulae are copied around a spreadsheet (a main source of errors - see 'Relative and Absolute Copy problems' in 6.2 Errors In Enabling Skills).

All these concepts were considered useful for an improved methodology. The methodology devised and tested at the University was based on:

1. Verifying a formula using the 3 A's : Appropriate, Accurate, AboutRight.
2. Verifying a formula using a cell note that describes the formula according to everyday language ie. according to the underlying data model.
3. Displaying in colour the extent to which a formula is copied from an originating cell to a range of receiving cells.

The methodology consisted of 5 steps.

Step 1 : Appropriateness

This is an aid to Planning Skills.

Appropriateness is the correctness of the formula according to the underlying data model of the business process being modelled. In Figure 4 one may ask what is meant by Average Pay in column F and can this be calculated as Total Wages divided by the Staff Numbers for that grade of staff . Also, one may ask 'Does it make business sense to do this - is this how one calculates average wage per staff member in a grade ?'

Figure 4

	A	B	C	D	E	F
1	Lazy Days Staff Budget Costs 1995-1996					
2		Staff	Day Work	Night Work	Total	Average
3		Num bers	Wages £	Wages £	Wages £	Wage £
4						
5	Managers	1	17700.50	0.00	17700.50	17700.50
6	Grade 1	3	45540.00	1400.55	46940.55	
7	Grade 2	9	122340.00	2000.00	124340.00	
8	Grade 3	12	102350.25	0.00	102350.25	
9	Grand Totals	25	287930.75	3400.55	291331.30	
10						

The spreadsheet builder can verify appropriateness by entering the real-world description of the formula in the cell note for the cell. The cell F5 cell note should contain the description as shown in Figure 5.

Figure 5

The screenshot shows a spreadsheet with a 'Cell Note' dialog box open for cell F5. The spreadsheet data is as follows:

	A	B	C	D	E	F	G
1	Lazy Days Staff Budget Costs 1995-1996						
2		Staff	Day Work	Night Work	Total	Average	Night
3		Numbers	Wages £	Wages £	Wages £	Wage £	Wages %
4							of Total
5	Managers	1	17700.50	0.00	17700.50	17700.50	

The 'Cell Note' dialog box shows:

- Cell: F5
- Text Note: Average Wage is Total wages for the grade divided by the number of staff in that grade.
- Buttons: OK, Close

Step 2 : Accuracy

This is an aid to Enabling Skills.

In this step the builder needs to check that the formula E5/B5 is the correct interpretation of the data model. In Figure 5 the formula in F5 is =E5/B5. In this step the builder needs to check that the formula E5/B5 is the correct interpretation of the data model description given in the cell note under the appropriateness check above. Questions needed to be asked here are:

are these the correct cell references? is the formula well-formed according to MS Excel rules and BODMAS rules ?

Step 3 : About Right

This is a general Planning Skill aid.

Is the resulting cell value in F5 about right numerically ? i.e. does 17700.50 look roughly correct ?

Step 4 : Validating a Formula Copy To A Cell Range

This is an aid to Planning and Enabling Skills.

Suppose the user wished to copy the formula from F5 to the cell range F6.F9. In Figure 4, the formula = E5/B5 has been entered in cell F5 (it produces the data 17700.50). An end-user would copy the formula from F5 to the receiving cell range F6 to F9 and the copy would alter the cell references in the original formula so that in F6 the formula would be =E6/B6, in F7 it would be =E7/B7 and so on. For an auditor, checking the spreadsheet later, it is not obvious that the formula was first written and verified in F5 and then copied to the other cells. In order to verify the correctness of each formula in the range F5 to F9 the auditor would need to individually inspect each cell. This is a lengthy process which can be avoided by specifically showing the originating cell and the receiving range in the same colour font. Using this method the auditor could tell that F5 was the originator cell because it had an attached cell note (cells with cell notes are denoted by a red dot in the top right hand corner when displayed on screen - this dot does not show in printed copy) and that F6 to F9 were the receiving cells of the copy from F5 because the entire range F5 to F9 is coloured in blue.

Figure 6

	A	B	C	D	E	F
1	Lazy Days Staff Budget Costs 1995-1996					
2		Staff	Day Work	Night Work	Total	Average
3		Num bers	Wages £	Wages £	Wages £	Wage £
4						
5	Managers	1	17700.50	0.00	17700.50	17700.50
6	Grade 1	3	45540.00	1400.55	46940.55	15646.85
7	Grade 2	9	122340.00	2000.00	124340.00	13815.56
8	Grade 3	12	102350.25	0.00	102350.25	8529.19
9	Grand Totals	25	287930.75	3400.55	291331.30	11653.25

Having validated the originator cell F5 by reading the cell note, the auditor has no need to inspect formulae in the receiving cells as they must all contain relative copies of the F5 formula.

The auditor can also see that the formula in F9 (Totals for column F) must be appropriate because, as it is a copy of the =E5/B5 in F5 then it must be =E9/B9 which is the appropriate calculation for calculating the average wage of all staff members. Contrast this with the situation in Figure 7.

Figure 7

	A	B	C	D	E	F
1	Lazy Days Staff Budget Costs 1995-1996					
2		Staff	Day Work	Night Work	Total	Average
3		Num bers	Wages £	Wages £	Wages £	Wage £
4						
5	Managers	1	17700.50	0.00	17700.50	17700.50
6	Grade 1	3	45540.00	1400.55	46940.55	15646.85
7	Grade 2	9	122340.00	2000.00	124340.00	13815.56
8	Grade 3	12	102350.25	0.00	102350.25	8529.19
9	Grand Totals	25	287930.75	3400.55	291331.30	55692.09

In Figure 7 the auditor it can be seen at a glance that the spreadsheet builder copied a formula from C9 (cell note attached) to D9.F9 (because all cell values in the range C9 to F9 are coloured red). He/she can also see that F5 was copied to F6.F8 (because all cell values in the range F5 to F8 are coloured in blue). If the formula in C9 was =SUM(C5.C8) then a relative copy of this has appeared in the receiving range. But the formula =SUM(F5.F8) would not be appropriate in F9 as totalling a list of averages does not give the average of the original totals - an About-Right check shows that the figure of 55692.09 cannot possibly be the average wage of all staff if the highest paid staff member, the manager, is only paid 17700.50 . An auditor could tell this just by looking at the colour codes for the formula copying.

Step 5 : Modularisation

This is an aid to Planning Skills.

The spreadsheet should be broken down into separate logical areas - each area should be prepared individually and bordered in bold lines on the screen display. Each 'module' could have as many colour codings of formula copies as necessary as each will be treated as a stand-alone piece. The tagged cells (those with cell notes) in a module must be checked individually but for each one, the auditor only has to compare the formula in the cell with the cell note description. He/she can then inspect the colour codings to validate that the formula was correctly copied to the correct receiving range and is appropriate for this range. Data dependencies between modules could then be checked using the Microsoft Excel audit tool.

See Figure 8 for a complete spreadsheet modularised with colour-coded formulae copying in use.

8.2 Testing of the Proposed Method

The 3 A's approach was tested by comparing two cohorts of students taught in different years:

- Students from a 1996 cohort had four supervised one hour practical sessions and four one hour formal lectures on spreadsheets. They were given the same test as the later 1997 cohort but were allowed to construct the spreadsheet first on computer. The test was not multiple choice - students were asked to write down the precise formula for a given cell.
- Students from the 1997 cohort had the same number of hours of supervised practicals as the 1996 cohort but with only a one hour formal spreadsheet lecture in total. Also, for the test, they were not permitted to construct the spreadsheet on computer first but were given the spreadsheet test on paper. This time, the paper was multiple choice with 6 choices per question.

One might reason that the 1996 cohort would outperform the 1997 simply because they had been permitted to build the spreadsheet on a computer before being questioned and because they had three more hours formal lecturing. However, the students in 1997 outperformed the 1996 cohort. They appeared to show much greater understanding of spreadsheet construction.

The difference between the two years was as follows. In 1997 the teaching approach was completely different. The teaching of spreadsheet skills were not taught as a separate item - they were taught as part of an integrated unit entitled Systems Building which taught Planning Skills and Enabling Skills together. It integrated business understanding with systems analysis and with software engineering techniques using case studies and scenarios that guided the student from beginning to end of a system development. This included spreadsheet development using the 3 A's Error-Checking approach which aided students to relate the spreadsheet model to its underlying data model. The integrating of Planning Skills with Enabling Skills gave students a much greater understanding of what they were doing. Lastly, an appreciation of the common errors that occur in industry helped the students to 'avoid doing things incorrectly'.

In addition to the above, five students were asked to construct a spreadsheet using the colour-coding of copied ranges and to report on how useful they found the technique. All reported positively on the methodology. The colour coding technique was found to be useful for lecturers marking students spreadsheet work as it enabled the lecturer to see the data, formulae and relations between copied formulae on one spreadsheet - something which was not possible before. In many cases, the lecturer could tell, almost at a glance, whether a spreadsheet was correct by looking at its colour-coding.

Figure 8 shows a spreadsheet in which colour codings of copy actions have been done in each module:

in the first module (Budget 1995-96)

- figures in F5.F8 are blue as they are copies of the F5 formula E5/B5
- figures in E5.E8 are green as they are copies of the E5 formula C5+D5
- figures in C9.F9 are red as they are copies of the C9 formula SUM(C5.C8); it is here that an auditor will spot the error in F9 as F9 should not contain a relative copy of the C9 formula as it is not appropriate to sum the values in F5.F8 (the average wage of each staff category) to determine the Average wage of all staff; instead, F9 should contain a copy of F5 formula;

in the second module (Budget 1996-97)

- figures in C21.D24 are coloured scarlet as they are all copies of the C21 formula C5+(C5*\$B13/100). Note that the \$ makes the B13 reference an absolute reference with respect to the column B)
- figures in E21.E24 are in green as all copies of E21 formula
- figures in F21.F25 are in purple as all copies of F21 formula
- figures in C25.E25 are in red as all copies of the C25 formula : this time there is no error in F25 as it does not contain a copy of the C25 formula but a copy of the F21 formula;

in the third module (Percentage Breakdowns for 1996-1997) formulae are copied down the relevant columns with each column a different colour.

Figure 8

	A	B	C	D	E	F
1	Lazy Days Staff Budget Costs 1995-1996					
2		Staff	Day Work	Night Work	Total	Average
3		Numb.	Wages £	Wages £	Wages £	Wage £
4						
5	Managers	1	17700.50	0.00	17700.50	17700.50
6	Grade 1	3	45540.00	1400.55	46940.55	15646.85
7	Grade 2	9	122340.00	2000.00	124340.00	13815.56
8	Grade 3	12	102350.25	0.00	102350.25	8529.19
9	Grand Totals	25	287930.75	3400.55	291331.30	55692.09
10						
11	Lazy Days Staff Budget Costs 1996 - 1997					
12	Percentage wage Increases					
13	Managers	2				
14	Grade 1	1.5				
15	Grade 2	1.5				
16	Grade 3	1				
17						
18		Staff	Day Work	Night Work	Total	Average
19		Nos	Wages £	Wages £	Wages £	Wage £
20						
21	Managers	1	18054.51	0.00	18054.51	18054.51
22	Grade 1	3	46223.10	1421.56	47644.66	15881.55
23	Grade 2	9	124175.10	2030.00	126205.10	14022.79
24	Grade 3	12	103373.75	0.00	103373.75	8614.48
25	Grand Totals	25	291826.46	3451.56	295278.02	11811.12
26						
27	Percentage Breakdowns for '96-'97					
28		Staff	Basic	Night	Total	
29		Nos %	Wages %	Wages %	Wages %	
30		of Numbers	of Basic	of Night	of	
31		Grand Total	Grand Total	Grand Total	Grand Total	
32	Managers	4.00	6.19	0.00	6.11	
33	Grade 1	12.00	15.84	41.19	16.14	
34	Grade 2	36.00	42.55	58.81	42.74	
35	Grade 3	48.00	35.42	0.00	35.01	
36	Totals	100.00	100.00	100.00	100.00	

9 CONCLUSION

The research appears to have the following findings:

- similar errors are committed by undergraduate students and endusers in industry;
- both students and endusers could benefit from a spreadsheet methodology;
- there is no generally well-known spreadsheet methodology available to lecturers and their students in academia nor for endusers and auditors in industry;
- a spreadsheet methodology has been subjected to trials and appears to confirm that students do benefit by being given a structured approach to creating spreadsheets particularly if planning skills are taught alongside enabling skills and students are made aware of the common errors that they are likely to make.

As the error experiences of students and end-users appeared to be similar, it is possible that the methodology described herein may be useful in industry.

10 REFERENCES

- Batra D. & Sein M.K. (1994) Improving Conceptual database design through feedback, *International Journal of Human-Computer Studies*, Vol 40, p653-676
- Chambers A.D. and Court J.M. (1994) *Computer Auditing 3rd Edition*. Pitman.
- Coates J., Rickwood C., Stacey R. (1989) *Control and Audit in Management Accounting*. Heinemann Professional Publishing.
- Davis S.J. (1996) Tools for Spreadsheet Auditing, *International Journal of Human-Computer Studies*, Vol 45, p429-442
- DiAntonio A.E (1986) *Spreadsheet Applications*. Prentice-Hall.
- Hendry D.G and Green T.R.G (1994) A Cognitive Interpretation of what Users think of the Spreadsheet Model, *International Journal of Human-Computer Studies*, Vol 40, p1033-1065.
- Kavanagh J. (1997) Shoddy Business Models Breed Financial Disaster, *Computer Weekly*, 19th June.
- Microsoft Corporation (1994) *Microsoft EXCEL users Guide*.
- Van Vliet P., Kletke M.G and Chakraborty G. (1994) The Measurement of Computer Literacy, *International Journal of Human-Computer Studies*, Vol 40, p835-857.