16 A Revised Classification of Spreadsheet Errors

Kamalasen Rajalingham,
University of Westminster, 115 New Cavendish Street, London W1W 6UW,
K.Rajalingham@wmin.ac.uk

ABSTRACT

This paper describes an improved and revised framework for the systematic classification of spreadsheet errors. In past publications, the derivation of the taxonomic scheme and the justification for the proposed approach were not discussed in adequate detail. The new revised classification addresses these limitations and presents clearer guidance on the classification of specific types of spreadsheet errors. Like the previous classification schemes, this revised taxonomy of errors is also aimed at facilitating a more thorough analysis and understanding of the different types of spreadsheet errors. It is more comprehensive than any presented or published before. Every class or category of errors is explained and supported by appropriate examples. The ability to place an error within a certain class in the taxonomy should enable us to understand other similar errors and devise a strategy to prevent their occurrence.

1. INTRODUCTION

As mentioned in previous publications [Rajalingham, 1999, 1999a, 2002; Chadwick, 1999], the phenomenon and magnitude of spreadsheet errors can be viewed or investigated from three distinct perspectives: frequency of spreadsheet errors, real-life consequences of these errors, and the occurrence of specific types of spreadsheet errors. This paper is mainly related to the third perspective, concentrating on the examination and classification of specific errors. This area has gained very little attention in the past, resulting in a lack of understanding of the nature and causes of spreadsheet errors. An analysis of specific types of errors should precede the development of strategies and solutions to deal with the problem effectively. This involves the classification of these errors.

The previous version of the taxonomy of spreadsheet errors was presented by Rajalingham et al [Rajalingham, 2000a]. This paper presents a more comprehensive classification of spreadsheet errors than ever presented or published before, following a meticulous analysis of the nature and characteristics of specific types of spreadsheet errors from a wide variety of sources. This classification is confined to only *user-generated spreadsheet errors*, as opposed to the occurrence of errors produced by the spreadsheet software, which is beyond the scope of the current research. The classification is based on a rational taxonomic scheme and is supported by a selection of generic and specific examples. Earlier versions of the taxonomy have been published [Rajalingham, 1998, 1999, 1999a, 2000].

2. THE CONCEPT OF TAXONOMY OR CLASSIFICATION

The concept of taxonomy or classification has been discussed in some detail by Rajalingham et al [Rajalingham, 2000a], mainly quoting Britannica.com [Britannica.com, 1999, 2000]. According to these sources, taxonomy refers to the science of classification, which is usually applied to the classification of living and extinct organisms. However, there is no special theory behind modern taxonomic methods [Britannica.com, 1999, 2000].

According to Britannica.com [Britannica.com, 1999, 2000], in biology, *taxonomy* refers to the establishment of a hierarchical system of categories on the basis of presumed natural relationships among organisms. The same source further states that the goal of classifying is to place an organism into an existing group or to create a new group for it. Rajalingham et al [Rajalingham, 2000a] adopted these definitions and extended the concept of taxonomy to the classification of spreadsheet errors. The *spreadsheet error taxonomy* can be defined as a hierarchical system of classes of spreadsheet errors on the basis of common characteristics and relationships.

3. RATIONALE FOR THE CLASSIFICATION OF SPREADSHEET ERRORS

There are various reasons for developing a classification of spreadsheet errors. It is a methodical approach to problem analysis. The analysis of the different types of errors based on this approach is likely to improve comprehensive testing of a spreadsheet development methodology. The classification of spreadsheet errors also enables us to gain a deeper understanding of the characteristics of an error as well as the nature of its occurrence. A comparison can also be made with other related errors belonging to the same class.

An insight into the characteristics and nature of an error is extremely important, in order to prevent the occurrence of the error or develop a method for detecting its presence. The identification of similar characteristics and properties between errors, may enable the development of similar approaches to deal with spreadsheet errors within the same taxonomic group. Knowledge of the characteristics of an error can also help in evaluating its potential impact and frequency, probably shared by other errors in the same category.

4. DERIVATION OF THE TAXONOMIC SCHEME

As there is no special theory behind modern taxonomic methods, methods of classification employed in other fields can be used to guide the process of classifying spreadsheet errors. Based on the principles of classification adopted in *zoology* and *botany* [Britannica.com, 1999, 2000], spreadsheet errors can be classified using a similar taxonomic scheme, consisting of the following steps:

- A specific type and example of a spreadsheet error is obtained.
- The error is compared with the known range of variation of spreadsheet errors.
- The error is correctly identified if it has been described, or a description showing similarities to and differences from known categories, is prepared. If the error is of a new type, it is assigned to a new category or class.
- The best position for the error is determined in the existing classification. This may also involve determining what revision the classification requires as a consequence of the new discovery.

- Available evidence is used to further suggest and describe the nature of the error, its possible causes and other characteristics.

Based on Britannica.com [Britannica.com, 1999, 2000], it is clear that the process of spreadsheet error classification requires a recognised system of ranks, rules and a verification procedure. An investigation of a taxonomic method that addresses these requirements revealed that there are two possible approaches to structuring the ranks within a taxonomy, a *binary* approach or a *bushy* approach. Both methods are based on a top-down approach that produces a hierarchical taxonomy, by studying the nature and characteristics of errors.

The bushy approach was initially adopted and assessed. A category at any level or rank can be divided into two or more classes. An example of an earlier version of the proposed taxonomy using the bushy approach is shown in *Figure 1* [Rajalingham, 1999, 1999a, 2000; Chadwick, 1999]. The *bushy* taxonomic structure shown in *Figure 1* was found to have certain limitations. It was difficult to navigate down the taxonomic tree to assign a specific error to a class. It was also possible to place certain errors in two or more different classes, potentially resulting in an ambiguous interpretation of the errors.

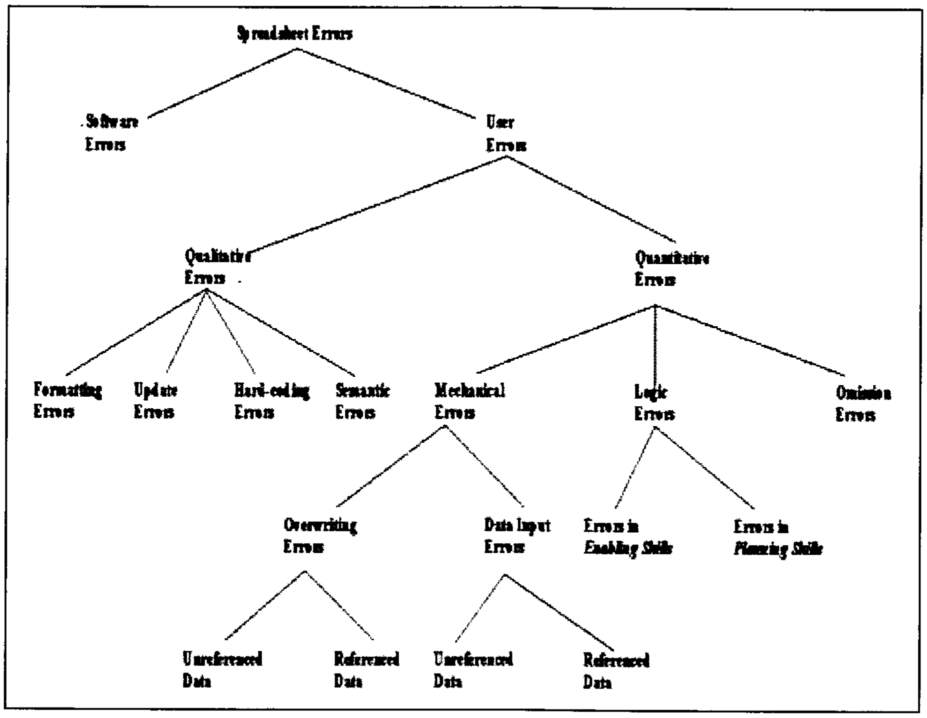


Figure 1: Taxonomy Using a *Bushy* Approach

In order to address these limitations, the alternative *binary* approach was considered. At each stage of the taxonomy, the binary approach uses *dichotomies* or division into two mutually exclusive *(non-overlapping)* groups, to classify the errors. This minimises the possibility of positioning the same type of error in different classes/sub-classes and causing an overlap. This feature of the binary approach makes it a far more straight-forward way of assigning a specific error to a taxonomic class. A simple IF-THEN-ELSE rule or constraint can be used to navigate down the taxonomy tree and position errors in appropriate classes. In order to reduce ambiguity, for each dichotomy, only a single factor, representing a distinct aspect of the error, is considered. To this end, the following aspects of a particular type of spreadsheet error are analysed:

- Manifestation of the error
- Cause of the error
- The role of the person responsible for the error
- The cognitive state of the person responsible for the error
- The stage of the spreadsheet building life cycle where the error occurs
- The relevant view of the spreadsheet model system

In view of the advantages of the *binary* method compared to the *bushy* method, the binary approach has been adopted as the basis of a rational taxonomic scheme for classifying spreadsheet errors.

5. THE CLASSIFICATION OF SPREADSHEET ERRORS

The term *error* used in this paper has a broader definition encompassing both *actual errors* and *potential errors*. The errors include *flaws*, *slips* and *mistakes*. In the process of classifying certain specific errors, assumptions are made about the precise cause of the errors, where this is not clearly described by the source. Otherwise, it would be possible to assign the same error to several different categories.

Based on the new revised classification of user-generated spreadsheet errors, at the highest level, spreadsheet errors can be divided into two non-overlapping categories of *quantitative* and *qualitative* errors. The classification factor used at this stage is the manifestation of the error. Panko and Halverson [Panko, 1996] have also broadly split spreadsheet errors into quantitative or qualitative errors.

For all *user-generated* spreadsheet errors:

IF numerical error causing incorrect bottom-line value
THEN ***quantitative*** error
ELSE *NOT quantitative* error (i.e. ***qualitative*** error)

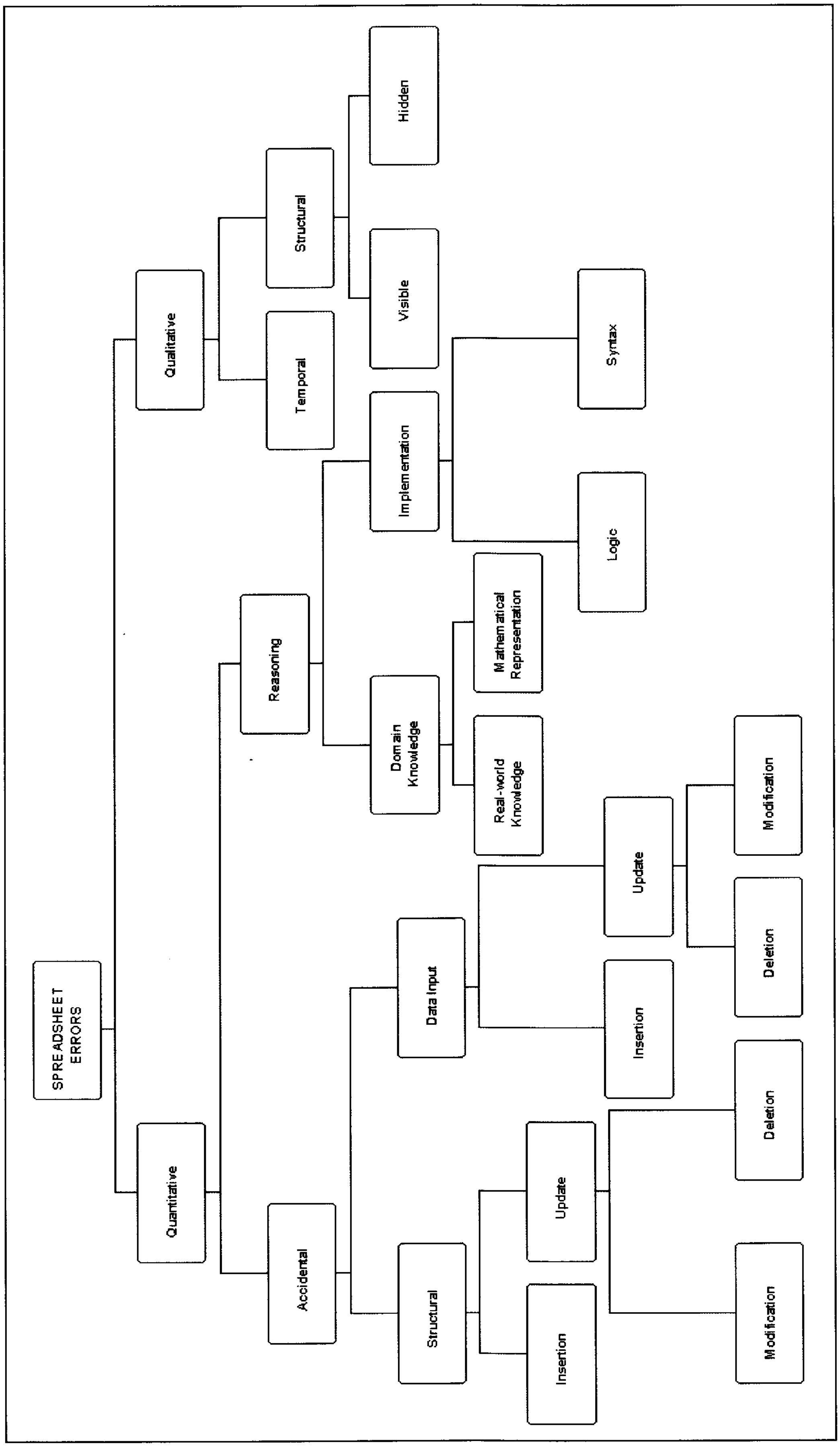
5.1 Quantitative Errors

Quantitative errors are numerical errors that lead to incorrect bottom-line values [Panko, 1996]. They simply produce wrong data in the spreadsheet model. Based on an analysis of the cause of the error, a dichotomy of *accidental* and *reasoning errors* can be used to capture the different types of quantitative errors. Any error or flaw, which is quantitative and not accidental, must have been produced as a result of a mistake in reasoning.

For all *quantitative* errors:

IF error is caused by negligence or carelessness
THEN ***accidental*** error
ELSE *NOT accidental* error (i.e. ***reasoning*** error)

The dimension of fraud is not taken into account when developing the classification framework for quantitative errors, as any error can be deliberately produced with fraudulent or malicious intent and disguised as an accidental or reasoning error.


SPREADSHEET ERRORS
Qualitative
Quantitative
Temporal
Structural
Visible
Hidden
Reasoning
Accidental
Implementation
Syntax
Logic
Domain Knowledge
Mathematical Representation
Real-world Knowledge
Data Input
Structural
Insertion
Update
Deletion
Modification
Insertion
Update
Deletion
Modification

1. Accidental Errors

Accidental errors are mistakes and slips caused by negligence, such as typographical or pointing errors. Though quite frequently occurring, they have a high chance of being spotted and corrected immediately. Based on the user role responsible for the error, an accidental error can either be a *structural error* or a *data input error*.

For all *accidental* errors:

IF error is caused by the model developer
THEN ***structural*** error
ELSE *NOT structural* error (i.e. ***data input***)

(a) Structural Errors

Structural errors are errors produced by the developer of the spreadsheet model. These errors are produced when creating or altering the structural or programmed component (formula network) of the spreadsheet model. Therefore, these errors can be further segregated into two categories, namely, *insertion* and *update* errors.

For all *structural* errors:

IF error is produced when creating the structural aspects of the spreadsheet model
THEN ***insertion*** error
ELSE *NOT insertion* error (i.e. ***update*** error)

(i) Insertion Errors

These errors occur while the developer is creating the structures of the spreadsheet model. The model would be prone to accidental errors such as typographical errors, pointing errors, duplication and omissions.

Example 1: Omissions

Omissions are important factors or variables that are left out of a spreadsheet model [Cragg, 1993]. According to Panko and Halverson [Panko, 1996], research had shown that omission errors were dangerous due to the low detection rates.

Example 2: Pointing Errors

Pointing errors refer to errors caused by references being made to wrong cells or cells in the wrong location. The model developer types the wrong cell coordinates in composing the formula [Brown, 1987]. As a result, the formulae themselves produce incorrect results and may even refer to blank cells or non-numeric cells.

(ii) Update Errors

These errors occur while the developer is altering the structural or programmed component (formula network) of the spreadsheet model. The model at this stage would be prone to accidental errors such as typing errors, overwriting and deletion.

For all *structural update* errors:

IF If error is produced as a result of incorrectly changing the structural or programmed component of the model
THEN ***modification*** error
ELSE *NOT modification* error (i.e. ***deletion*** error)

Modification Errors

These errors occur as a consequence of incorrectly or inaccurately modifying the structural or programmed component (formulae) of the spreadsheet model. The modification of spreadsheets is more prone to errors compared to the original creation of spreadsheets [Brown, 1987].

Example 1: Formulae Overwritten with Data

Data is incorrectly entered into a cell previously containing a formula, overwriting the formula and invalidating the model [Cragg, 1993]. Stang [Stang, 1987] and Hayen [Hayen, 1989] also described this error. A simple solution to the problem is to use cell protection.

Example 2: Formula Overwritten with an Incorrect Formula

Similar to the previous example. However, the correct formula is accidentally replaced with an erroneous formula.

Deletion Errors

These update errors, on the other hand, are produced as a result of deleting or erasing existing elements of the structural or programmed component of the spreadsheet model.

Example 1: Deletion of a Formula

A correct formula required by the spreadsheet model is accidentally erased. The main cause of this error is the failure to protect cells containing formulae.

(b) Data Input Errors

Data input errors are errors made by end-users who merely manipulate the spreadsheet model. They are caused by erroneous entry of data required by the model. These errors can occur while either inserting new data or amending/updating existing data.

For all *data input* errors:

IF error occurs when entering new data into the spreadsheet model
THEN ***insertion*** error
ELSE *NOT insertion* error (i.e. ***update*** error)

(i) Insertion Errors

These errors are produced while entering new data into the model. Typically these would take the form of typographical errors or omissions committed by the data entry users.

Example 1: Erroneous Data Input

Invalid or incorrect data is easily entered into the spreadsheet model because there are no data checks on entry. Sometimes, the right data is put in the wrong cell. Wrong data can occur either due to a data entry error or incorrect data held by the data source [Hayen, 1989]. Freeman [Freeman, 1996] proposes the use of limit controls (tolerable ranges) to deal with these errors.

Example 2: Omissions

It is not uncommon for data entry operators to accidentally leave out certain inputs to the model.

(ii) Update Errors

These errors are produced as a result of incorrectly updating existing data in the model. Update operations (apart from insertion) must either be modification (or overwriting) or deletion.

For all *update* errors:

IF error occurs as a result of overwriting existing data
THEN ***modification*** error
ELSE *NOT modification* error (i.e. ***deletion*** error)

Modification Errors

These errors are produced as a result of changing existing data in the model. Typically these would be typographical or overwriting errors committed during data entry.

Example 1: Overwriting of Data

A correct piece of data entered is overwritten with an incorrect input. This might be caused by an update being applied in the wrong location.

Deletion Errors

These errors, on the other hand, occur as a result of deleting or erasing previously entered data from the model. These errors are also caused by users responsible for data entry.

Example 1: Erasure of Data

A correct piece of input required by the model is simply deleted inadvertently. This is usually done during data entry or update.

2. Reasoning Errors

Reasoning errors are mistakes in reasoning and therefore not accidental. They are produced as a result of a lack of knowledge required to comprehend, analyse and accurately model the business function or problem in the form of a spreadsheet model. Reasoning errors can be split into two distinct categories based on an analysis of the precise cause of the errors, which in this case also involves a study of the cognisance of the model developer. The two mutually exclusive classes of reasoning errors are *domain knowledge errors* and *implementation errors*.

Any reasoning error which is not produced owing to inadequate comprehension of the underlying problem or function to be modelled on the spreadsheet, could only possibly have been caused by an incorrect implementation of the problem or function using the spreadsheet package. Implementation errors are far more common than domain knowledge errors, though domain knowledge errors are generally more serious.

For all *reasoning* errors:

IF error occurs owing to a lack of understanding of the underlying problem or function to be modelled
THEN ***domain knowledge*** error
ELSE *NOT domain knowledge* error (i.e. ***implementation*** error)

(a) Domain Knowledge Errors

Domain knowledge errors are specifically caused by inadequate awareness or knowledge required to identify, analyse and understand the business function or problem underlying the spreadsheet model. This knowledge is essential for modelling the problem and designing the corresponding conceptual or logical data model.

This category of errors consists of two distinct classes, namely *real-world knowledge* and *mathematical representation* based errors. Any reasoning domain-knowledge error which occurs despite selection of the right algorithm must have been caused by a lack of understanding of how the algorithm is to be mathematically represented.

For all *domain-knowledge* errors:

IF error caused as a consequence of a lack of knowledge on the underlying algorithm of a calculation or function
THEN ***real-world knowledge*** error
ELSE *NOT real-world knowledge* error
(i.e. ***mathematical representation*** error)

(i) Real-world Knowledge Errors

These errors involve creating a formula by selecting the wrong algorithm. Users may select an inappropriate template for a particular analysis or decision task, due to a lack of accounting knowledge or intellectual modelling logic.

Example 1: Exclusion of Factors from Formulae

A fairly common error in this category is the exclusion of important factors in a calculation. For instance, *bad debt provision* is excluded in an accounting calculation.

Example 2: Absence of Distinction Between Leap and Non-leap Years

This is a simple example of a real-world knowledge error whereby to calculate the daily figures for a particular leap year, the calculations divide by **365** instead of **366**.

(ii) Mathematical Representation Errors

These errors involve constructing the wrong formula despite having selected the right algorithm. This is due to a lack of knowledge on how to represent a mathematically correct and accurate formula based on the correctly chosen algorithm.

Example 1: The PERCENTAGE Error

This error occurs when the formula to calculate percentage is incorrectly written, due to a lack of knowledge of how to calculate a percentage or BODMAS (Brackets, Of, Division, Multiplication, Addition, Subtraction), which identifies precedence in calculations.

Example 2: Incorrect Representation of an OVERALL AVERAGE Function

Based on *Figure 2*, the correct formula in **F9** is **=E9/B9** but the formula **=AVERAGE(F5:F8)** is entered instead [Chadwick, 1997, 1997a]. Although the model developer knew that an overall average was to be calculated, they incorrectly assumed that the sum of averages would give the overall average.

	A	B	C	D	E	F
1	Lazy Days Staff Budget Costs 1995-1996					
2 3		Staff Numbers	Basic Wages £	Overtime Wages £	Total Wages £	Average Wage £
4						
5	Managers	1	17700	0		
6	Grade 1	3	45540	1400		
7	Grade 2	9	122340	2000		
8	Grade 3	12	102350	0		
9	Grand Totals	25	287930	3400		

Figure 2: Example to Illustrate *Overall Average* Error

(b) Implementation Errors

Implementation errors are produced due to a lack of knowledge or understanding of the full use of the functions and capabilities of the particular spreadsheet package in use, with an understanding of the spreadsheet principles, concepts, constructs, reserved words and syntax. Implementation errors consist of *logic* and *syntax* errors.

For all *implementation* errors:

IF error is caused by a lack of comprehension of the features and functions of the spreadsheet package/language
THEN ***logic*** error
ELSE *NOT logic* error (i.e. ***syntax*** error)

(i) Logic Errors

Logic errors are errors caused by a lack of understanding of the functions and capabilities of a specific spreadsheet package, which enable the accurate representation of a solution.

Example 1: RELATIVE and ABSOLUTE Copy Problem

The relative copy causes cell references in a copied formula to alter row and column references relative to the original cell copied [Chadwick, 1997). The error is also caused by copying a formula hidden underneath a cell value, thinking that it is the value that is being copied [Brown, 1987].

Example 2: Value Not Included in the Total

This error has been pointed out by several authors [Ayalew, 2000; Butler, 1997; Stang, 1987; Ditlea, 1987]. The modeller writes a formula to find a range total in cell **B10**. The formula is **=SUM(B1:B9)**, and data are entered in cells **B1 to B9**. A row is then inserted below cell **B9** and a new value entered in **B10**. This cell is beyond the range of the formula (which has now been shifted to **B11**) and therefore not included in the addition.

Example 3: Rounding Error

Rounding can and should always be controlled. The best approach is to perform all operations on rounded numbers, and not with "hidden" or formatted values. Based on *Figure 3* [Batson, 1991], it can be seen that the "formatted" column does not add up, and therefore affects the credibility of the model.

	Actual	Formatted	Rounded
A1	1.128431	1.13	1.13
A2	2.35625	2.36	2.36
A3	1.827994	1.83	1.83
=SUM(A1:A3)	5.312675	5.31	5.32

Figure 3: Rounding Error

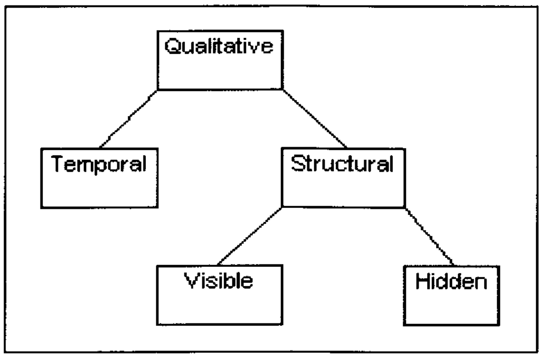
(ii) Syntax Errors

Syntax errors are errors caused by a lack of precise understanding of the constructs, reserved words and syntax of a specific spreadsheet package, used to write functions and formulae.

Example 1: A Keyword Within a Formula is Misspelled

A keyword within a formula is misspelled causing an error, e.g. =AVG(...) instead of =AVERAGE(...). This error can be easily detected.

5.2 Qualitative Errors



Qualitative errors are errors that do not immediately produce incorrect numeric values but degrade the quality of the model. The model also becomes more prone to misinterpretation, and difficult to update/maintain. Qualitative errors can be generally divided into two different types, namely, *temporal errors* and *structural errors*.

This dichotomy is obtained mainly based on an analysis of the three views of an information system: *data, processing* and *behaviour*. Within the context of spreadsheet models, the *processing view* of a model is the network of formulae used to perform calculations, while the *data view* represents the various input data required for the calculations. The *behavioural* or *temporal view* represents the effects of time and real world events on the spreadsheet model. A qualitative error which is not temporal can be considered a structural error. The structural aspect of the model represents the formula network and data.

For all *qualitative* errors:

IF error is caused by an elapse of time, which invalidates data
THEN ***temporal*** error
ELSE *NOT temporal* error (i.e. ***structural*** error)

1. Temporal Errors

Temporal errors are qualitative errors which invalidate data (and possibly formulae) with the passage of time. As a result, the model ceases to be reliable. Typically these errors are caused by failure or delays in updating the spreadsheet model to reflect current circumstances.

Example 1: Qualitative Error Caused by the Referencing of Non-current Data

This error is produced as a result of referencing a piece of data that has become invalid/inaccurate due to time lapse.

Example 2: A Previous Erroneous Model is Used

There may be different versions of a model, where each version may have been revised differently by some user [Stang, 1987]. It is possible that the most current version has been fully debugged, but a previous version with flaws is used to make important decisions.

2. Structural Errors/Flaws

Structural errors in this context can be defined as non-temporal qualitative errors or flaws produced as a result of poor design or layout of model structures and data. Based on the physical manifestation of these errors, they can be divided into two categories: *visible errors* and *hidden errors*.

For all *structural* errors:

IF error is a structural flaw which is visible at the surface level of the model
THEN ***visible*** error
ELSE *NOT visible* error (i.e. ***hidden*** error)

(a) Visible Errors/Flaws

Visible qualitative structural errors are structural flaws which are visible at the surface level of the model. The detection of these errors does not require any examination of the formula level. These errors normally take the form of semantic errors which make the models more prone to misreading or misinterpretation.

Example 1: Formatting Error

A common qualitative error is where the cell format is specified as *general* on the spreadsheet. Consequently, the figures have varying decimal places and make it difficult to identify a number that is incorrect. This is shown in *Figure 4*. The value in cell **G10** is greater than the value in cell **G9**. However, at a quick glance, the value in cell **G9 (102350.25)** may seem to be greater than the value in cell **G10 (291331.3)** due to the inconsistent use of decimal places [Chadwick, 1997].

C	D	E	F	G	H	
	Number of Staff	Day Wages £	Night Wages £	Total Wages £	Average Wage £	5
Grade 1	1	17700.5	0	17700.5	17700.5	6
Grade 2	3	45540	1400.55	46940.55	15646.85	7
Grade 3	9	122340	2000	124340	13815.56	8
Grade 4	12	102350.25	0	102350.25	8529.19	9
Grand Total	25	287930.75	3400.55	291331.3	11653.25	10

Figure 4: Formatting Error

(b) Hidden Errors/Flaws

Hidden qualitative structural errors, on the contrary, are structural flaws which are not visible at the surface level of the model and therefore require examination of the formula level. These errors normally take the form of complicated, confusing or inappropriate construction of formulae. Such flaws can make the model difficult to maintain and prone to inconsistencies or update anomalies.

Example 1: Hard-coding

A fixed value is used when a variable (cell reference) should be used instead. In other words, the cell contains a hard-coded input instead of a formula. For instance, *net * 17.5%* instead of *net * a named variable or range for VAT rate* [Butler, 2000].

Example 2: Complexity of Formulae

Stang [Stang, 1987] states that any equation longer than 80 characters uses logic that is difficult to follow.

6. CONCLUSION

In spite of an increasing awareness of the consequences of user-generated spreadsheet errors, there has been a lack of research and analysis of specific types of these errors. In order to effectively deal with the problem, a thorough examination and classification of specific types of spreadsheet errors is essential.

This paper has described a more comprehensive taxonomy of user-generated spreadsheet errors than ever presented or published before, based on a rational taxonomic scheme. The systematic classification of spreadsheet errors enables us to gain a far better understanding of the different types of errors. This facilitates the development of tools, techniques and methods to prevent their occurrence or improve the detection of existing errors.

7. REFERENCES

Ayalew, Y., Clermont, M. and Mittermeir, R.T. (2000) "Detecting errors in spreadsheets". In: Chadwick, D. (ed.) (2000) *EuSpRIG 2000 Symposium proceedings - spreadsheet risks, audit and development methods*. London: University of Greenwich, pp. 51-62.

Batson, J. and Brown, A. (1991) "Spreadsheet modelling best practice", *Accountants Digest*, 2730 (Autumn).

Britannica.com (1999-2000) *Website: http://www.britannica.com*. Britannica.com and Encyclopaedia Britannica, Inc.

Brown, P.S. and Gould, J.D. (1987) "An experimental study of people creating spreadsheets", *ACM Transactions on Office Information Systems*, 5(3), pp. 258-272.

Butler, R. (1997) "The subversive spreadsheet", *DataWatch (ISACA London, UK)*.

Butler, R. (2000) "Risk assessssment for spreadsheet developments". In: Chadwick, D. (ed.) (2000) *EuSpRIG 2000 Symposium proceedings - spreadsheet risks, audit and development methods*. London: University of Greenwich, pp. 65-74.

Chadwick, D., Knight, J. and Clipsham, P. (1997) "Information integrity in end-user systems". In: Jajodia, S., List, W., McGregor, G. and Strous, L. (eds) (1997) *Integrity and internal control in information systems, volume 1: increasing the confidence in information systems*. London: Chapman and Hall, pp. 273-292.

Chadwick, D. (1997a) "Auditing and the three A's", *CASG Journal*, 7(4), p. 7.

Chadwick, D., Rajalingham, K., Knight, B. and Edwards, D. (1999) "A methodology for spreadsheet development based on data structure", *CMS Press*, 99/IM/50.

Cragg, P.G. and King, M. (1993) "Spreadsheet modelling abuse: an opportunity for OR?", *Journal of the Operational Research Society*, 44(8), pp. 743-752.

Ditlea, S. (1987) "Spreadsheets can be hazardous to your health", *Personal Computing*, 11, pp. 60-69.

Freeman, D. (1996) "How to make spreadsheets error-proof", *Journal of Accountancy*, May, pp. 75-77.

Hayen, R.L. and Peters, R.M. (1989) "How to ensure spreadsheet integrity", *Management Accounting*, 60(9), pp. 30-33.

Panko, R.R. and Halverson, R.P., Jr. (1996) "Spreadsheets on trial: a survey of research on spreadsheet risks", *Proceedings of the Twenty-Ninth Hawaii International Conference on Systems Sciences, 2-6 January 1996, Maui, Hawaii.* California: IEEE Computer Society.

Rajalingham, K. and Chadwick, D. (1998) "Integrity control of spreadsheets: organisation & tools". In: Jajodia, S., List, W., McGregor, G.W. and Strous, L. (eds) (1998) *Integrity and internal control in information systems.* Massachusetts: Kluwer Academic Publishers, pp. 147-168.

Rajalingham, K., Chadwick, D., Knight, B. and Edwards, D. (1999) "An approach to improving the quality of spreadsheet models". In: Hawkins, C., King, G., Ross, M. and Staples, G. (eds) (1999) *Software quality management VII – managing quality.* Great Britain: British Computer Society, pp. 117-131.

Rajalingham, K., Chadwick, D., Knight, B. and Edwards, D. (1999) "Efficient methods for checking integrity: an integrated spreadsheet engineering methodology (ISEM)". In: van Biene-Hershey, M.E. and Strous, L. (eds) (1999) *Integrity and internal control in information systems – strategic views on the need for control.* Massachusetts: Kluwer Academic Publishers, pp. 41-58.

Rajalingham, K., Chadwick, D., Knight, B. and Edwards, D. (2000) "Quality control in spreadsheets: a software engineering-based approach to spreadsheet development". In: Sprague, R.H., Jr. (ed.) (2000) *Proceedings of the Thirty-Third Annual Hawaii International Conference on System Sciences 2000 – abstracts and CD-ROM of full papers.* California: IEEE Computer Society.

Rajalingham, K., Chadwick, D. and Knight, B. (2000) "Classification of spreadsheet errors". In: Chadwick, D. (ed.) (2000) *EuSpRIG 2000 Symposium proceedings - spreadsheet risks, audit and development methods.* London: University of Greenwich, pp. 23-34.

Rajalingham, K., Chadwick, D. and Knight, B. (2002) "Efficient methods for checking integrity: a structured spreadsheet engineering methodology", *Informatica: An International Journal of Computing and Informatics*, 26(1).

Stang, D. (1987) "Spreadsheet disasters I have known", *Information Center*, 3(11), pp. 26-30.