The potential of Example Driven Modelling for Decision Support Spreadsheets

S. Thorne, D. Ball
University of Wales Institute Cardiff
SThorne@uwic.ac.uk, DBall@uwic.ac.uk

Abstract

In this paper we present a novel alternative means of modelling Decision support spreadsheets called Example Driven Modelling. The concept and rationale of the approach is discussed and experimental evidence is presented illustrating the benefits of the methodology. We demonstrate the application of EDM on a real world spreadsheet and compare the performance of the EDM model with the traditional spreadsheet. Finally we discuss the relative advantages and disadvantages of this approach and consider other areas of application in spreadsheet modelling.

1.0 Introduction

Spreadsheet error is evident in at least 30% of all spreadsheet models [1]. An example of the impact a spreadsheet error can have in industry is the loss of $24 Million by Trans Atlanta Corporation due to a copy and paste error when using a spreadsheet to bid for energy contracts in New York, USA [2]. Trans Atlanta Corporation is one example of many where spreadsheet errors cause significant financial loss in organisations.

Attempts to mitigate error include auditing methodologies, auditing software, development methodologies, best practice guidelines, testing strategies and alternative development or control environments. On the whole these different approaches solve parts of different problems and represent a growing body of research in spreadsheet error.

Presented in this paper is an alternative approach based upon the idea that spreadsheet errors are created through a poor interaction between human and computer.

1.1 Spreadsheet errors – A mismatch between man and machine?


Much of Michie’s work throughout the 1970’s to 90’s was concerned with Machine Learning Techniques (MLT) and the comparison of those techniques with equivalent human abilities [4 and 5]. He also revealed insights into the human learning process [6].

Michie argued that the human computer interaction was fundamentally limited due to the way in which humans interact with the computer. Michie essentially argued that the roles of machine and human in interaction did not exploit either’s strengths. In that vein, we argue that spreadsheet errors are mainly attributed poor interaction between humans and computers.

Consider the way in which a user interacts with a computer to create a model, such as a business problem on a spreadsheet, there are two fundamental processes.

The first element is matching patterns in real-world examples and realising trends in those patterns that form some rule or judgement. This allows the modeller to interpret and rationalise the problem and make rules that operate in that problem.

The second is the manipulation of mathematics and logic to represent that system accurately, which could be via a spreadsheet or another tool.

Now if we consider table 1, the natural strengths of the average human and the conventional computer, some discrepancies arise.
Table 1 Natural strengths of human and computer

<table>
<thead>
<tr>
<th></th>
<th>Human</th>
<th>Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern matching</td>
<td>Strong</td>
<td>Weak</td>
</tr>
<tr>
<td>Generating real world examples</td>
<td>Strong</td>
<td>Weak</td>
</tr>
<tr>
<td>Manipulating mathematics</td>
<td>?</td>
<td>Strong</td>
</tr>
<tr>
<td>Logical deduction</td>
<td>?</td>
<td>Strong</td>
</tr>
</tbody>
</table>

From this table we suggest that humans are strong at generating real-world examples and pattern matching but weak at mathematical manipulation and logical deduction. Conversely, computers are strong at manipulating mathematics and logical deduction but weak at generating real-world examples and pattern matching.

In the current spreadsheet development paradigm, strain is placed on the natural weakness of the human, logical deduction and manipulating mathematics, i.e. thinking up formulae to satisfy a problem (see the circled section on table 1). Further the strengths of the human (pattern matching and generating real-world examples) are not exploited by the current spreadsheet paradigm.

A potentially more beneficial paradigm would be to play on the natural strengths of the human and the conventional computer. In this new paradigm, the human would pattern match and generate real world examples, the computer would use its ability of mathematical manipulation and logical deduction to build a model from the examples provided by the user, see the circled sections in table 2.

Table 2 Proposed human spreadsheet interaction method

<table>
<thead>
<tr>
<th></th>
<th>Human</th>
<th>Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern matching</td>
<td>Strong</td>
<td>Weak</td>
</tr>
<tr>
<td>Generating real world examples</td>
<td>Strong</td>
<td>Weak</td>
</tr>
<tr>
<td>Manipulating mathematics</td>
<td>?</td>
<td>Strong</td>
</tr>
<tr>
<td>Logical deduction</td>
<td>?</td>
<td>Strong</td>
</tr>
</tbody>
</table>

1.2 Example-Driven Modelling

The inspiration for Example Driven Modelling (EDM) is realisation that most individuals "know what they mean" when it comes to creating representative models of a particular problem, i.e. a professional knows the rules that operate in their trade. However the problem arises upon implementing these rules since the rules then have to be translated into an acceptable syntax and the modeller must operate within the bounds of the software, this gives ample opportunity for the modeller to incorrectly program, misapply the logic of a rule or forget to include something.

EDM collects example attribute classifications, provided by the user, and uses these examples to generalise rules to apply to new unseen data.

To clarify, figure 1 shows the concept from start to end. Firstly the user would have to provide example attribute classifications for the problem they wish to model. The examples are then formatted into a data set and fed through a learning algorithm. The algorithm learns from the example data, provided which results in a general model, which is able to generalise to new unseen examples in the problem domain.

![Figure 1 Example-Driven Modelling (EDM)]
For example, consider a student grading system that computes the end of module result for a particular subject. The attributes present would be coursework and exam and the classification would be the resulting grade. So in order to provide attribute classifications of the grading system, examples need to be provided as follows:

<table>
<thead>
<tr>
<th>Coursework</th>
<th>Exam</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td>60</td>
<td>2:2</td>
</tr>
<tr>
<td>42</td>
<td>28</td>
<td>Fail</td>
</tr>
<tr>
<td>47</td>
<td>51</td>
<td>Pass</td>
</tr>
<tr>
<td>65</td>
<td>72</td>
<td>2:1</td>
</tr>
<tr>
<td>71</td>
<td>88</td>
<td>1st</td>
</tr>
</tbody>
</table>

The neural network then takes this input and generalises rules that apply to the attribute classifications provided. The resulting model can then be used on new unseen data.

This approach eliminates the need for the user to produce formulae, the user only gives example data for the problem they wish to model. This therefore eliminates errors in constructing formulae since the user is no longer required to produce them.

The burden of calculation is placed on the computer, which using a machine learning algorithm, computes the function of the examples. As the literature suggests, this may be a more effective use of human and computer strengths [3].

However in the case of example giving for EDM this is only theory and some investigation into the feasibility of such an approach is required, i.e. how feasible is it for humans to think up examples for a given problem.

### 2.0 Investigating the feasibility of giving examples

To investigate if example-giving works in practice an experiment was designed to compare traditional spreadsheet modelling techniques and the novel approach of giving examples. The first group, the “treatment” group, were required to give example data to complete the tasks. The other group, the control group, were given the same tasks to complete using a spreadsheet application.

The experiment into feasibility was designed in accordance guidelines cited by Shadish et al. [7] and Campbell and Stanley [8]. Also, published work using experimental methodologies in spreadsheet research were considered [9, 10, 11, 12 and 13].

### 2.2 Experiment aim

The main aim of the experiment was to establish experimentally within an academic environment, using postgraduate students:

1. The relationship between error and task complexity using a) spreadsheet modelling techniques, b) example giving
2. The (hypothesised) superiority of Example giving over traditional spreadsheet modelling.
3. A satisfactory statistical measure of overconfidence.
4. The relationship between previous spreadsheet experience and accuracy for both traditional spreadsheet modelling and example giving

From these aims and objectives, we will be able to determine the feasibility of Example giving via three performance indicators

1. Whether the participants understand the instruction of giving examples, i.e. can users understand the instructions of giving examples and generate valid examples in the context of the experiment tasks.
2. The accuracy of the examples provided by the participants, i.e. what is the error rate for examples provided by participants
3. The comparative error rate when compared to traditional modelling, i.e. how does the error rate compare to that of traditional modelling and does this warrant further investigation.

### 2.3 Experimental design

The experimental model chosen to evaluate the aims of the experiment is the “Randomised two-group no posttest design”. Figure 2 shows
the standard design of such experiments, this diagram is read from left to right and shows the

![Figure 2 Randomised two group no post test](image)

The diagram shows the two randomised (R) groups, the treatment group (X), the control group (which is left blank) and the two outcomes (O). In this case the control group receive ‘standard’ treatment, i.e. they develop spreadsheet formulae using the constructs and syntax in a spreadsheet application, such as Excel. The treatment group receive the novel approach, this allows relative comparison between the control and treatment groups.

2.4 Sampling

This sampling for this experiment was a cluster random sample as described by Shadish et al. [7] and Saunders et al. [14]. Cluster sampling identifies a suitable cluster of participants and then randomly selects from within that group.

Considering similar development experiments in [9, 10, 11, 12 and 13], postgraduate Masters students were selected as an appropriate cluster.

Selection within the cluster was random, participants were not divided upon ability or any other basis.

Participants were invited to attend a session arranged for the experiment. Upon arriving participants were divided into two groups, the control and treatment groups. The appropriate materials for each group were distributed and the experiment began.

2.5 Research materials

The research materials for this experiment comprise two different packs handed to the participants.

Both packs contained a questionnaire gathering information such as age, sex, experience, number of years using spreadsheets, and a personal rating of their skill. This questionnaire was completed first, before the participants started the tasks. The point of this questionnaire is to gather demographic information and to determine the experience of spreadsheet use for a participant.

Once questionnaire 1 was completed, the participants started the tasks for the group they were assigned to (control or treatment). The scenarios contained in tasks for the participants, regardless of group, were identical. The manner in which the groups completed the tasks differed, the control group produced formulae in a spreadsheet using the syntax and functionality of the application (Microsoft Excel). The treatment group produced example attribute classifications for each task.

After completing the tasks as best they could, the final questionnaire, questionnaire 2, was completed. This questionnaire gathered information on the participant’s perception of their own performance, i.e. they were asked how difficult they felt each task was and then asked to indicate how confident they were that the provided answers were correct.

2.6 Experimental tasks

The experiment tasks were designed to be progressively more difficult, requiring progressively more complex answers from both groups.

2.7 Control and Treatment tasks

The tasks given to the control and treatment group were identical, the method in which they answered varied. The control group submitted answers created using Microsoft Excel, the treatment group submitted attribute classifications written on paper.

For example, in the control group task 1 was to create a formula that could give a grade (Pass or Fail) based upon a single mark (Exam mark). The formula was required to distinguish between pass and fail, where fail was < 40 and pass was >= 40.

For the same task, task 1, the treatment group were required to give attribute classifications (examples) for every classification in the
problem. The two classifications are pass and fail, the participants therefore had to submit an attribute classification of pass and fail.

The tasks were also designed to be progressively more complex. For example task one uses one value (exam mark), 2 classifications (pass and fail) and two parameters for those classes (<40 Fail, >= 40 Pass).

In contrast, task 5 uses 2 values (exam and coursework mark), 4 classifications (fail, pass, merit and distinction) 4 parameters (< 40 fail, >= 40 pass, >= 55 merit and >= 70 distinction) and 1 conditional rule (Both exam and coursework values must fall in same class to award that class).

2.7.1 The experiment results

The summary statistics showed the treatment group were consistently more accurate in their tasks, i.e. those participants providing attribute classifications were more accurate at meeting the specification of the set problem. The control group were therefore less accurate at satisfying the set problems using traditional spreadsheet techniques. Figure 2 shows the performance of both groups in the five tasks.

![Figure 2 Accuracy of treatment and control groups](image)

2.7.2 Statistical significance of results

The summary statistics were then examined for statistical significance using chi squared, Fisher’s exact and Mcnemar’s test. The significance testing showed that the gain in accuracy experienced by the treatment group was only statistically significant in the final task (the most complex), i.e. the advantage in task 5 cannot be attributed to chance.

![Figure 3 Statistical significance of treatment and control group accuracy](image)

2.8 Application areas for EDM

As with all new thinking and novel ideas, EDM has a specific application to a relatively small number of spreadsheets, i.e. EDM is not a panacea.

For example EDM cannot be used to model balance sheets, profit and loss accounts and other such financial tools. However, it is applicable to decision support activities such as credit risk classification or medical risk classification.

One such example of a real world decision support spreadsheet is the Cardiac Anaesthesia Risk Evaluation (CARE) spreadsheet model. This spreadsheet is used to determine the risk of mortality, morbidity and a prolonged length of stay in hospital for cardiac patients.

3.0 The Cardiac Anaesthesia Risk Evaluation (CARE) algorithm

The CARE algorithm was created by Dupious and Wang [15] and is intended to be used by anaesthesiologists to determine risk of mortality, morbidity and a prolonged length of stay in hospital.

The inputs for CARE are: Cardiac disease (severity) (A), Number of controlled non-cardiac diseases (B), Number of uncontrolled non-cardiac diseases (B), Cardiac surgery (complexity) (C) and Urgency (emergency treatment or otherwise) (D). Table 4 describes
how each of the inputs (A to D) relate to each risk classification. The CARE algorithm uses 5 inputs and has 8 classifications.

<table>
<thead>
<tr>
<th>Input/ Risk Class</th>
<th>(A)</th>
<th>(B)</th>
<th>(C)</th>
<th>(D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(A1 &amp;</td>
<td>B1 &amp;</td>
<td>C1)</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>(A1 &amp;</td>
<td>B2 &amp;</td>
<td>C1)</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>(A2 V</td>
<td>B2 V</td>
<td>C2) &amp;D1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>(A2 V</td>
<td>B2 V</td>
<td>C2) &amp;D2</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>(A2 V</td>
<td>B3 V</td>
<td>C2) &amp;D1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>(A2 V</td>
<td>B3 V</td>
<td>C2) &amp;D2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>(A3 &amp;</td>
<td>-</td>
<td>C3 &amp;</td>
<td>D1)</td>
</tr>
<tr>
<td>8</td>
<td>(A3 &amp;</td>
<td>-</td>
<td>C3 &amp;</td>
<td>D2)</td>
</tr>
</tbody>
</table>

Table 3 CARE algorithm classifications and inputs

3.1 Problems with the CARE spreadsheet

There are a number of errors in the CARE spreadsheet, it is important to note that errors originate not from the medical algorithm but from the spreadsheet.

In summary the CARE spreadsheet has some serious issues:

1. The spreadsheet has multiple serious errors and examples of poor spreadsheet programming and design.
2. The CARE spreadsheet has difficulty classifying combinations of inputs that are ‘abnormal’, i.e. errors are present if certain combinations of valid input are entered.
3. The CARE spreadsheet has been programmed to explicitly test 8 of the possible 54 combinations of input.

3.2 Comparing the performance between the CARE spreadsheet and the EDM CARE model

Once the CARE EDM model was generated, a test set of unusual input was passed through both the EDM model and the spreadsheet model.

Unusual input consists of extraneous input values designed to test the limits of each model.

With “normal testing” both methods perform well.

Table 5 shows the results of this test, in the “model output” section correct output is expressed with a “✓” symbol, incorrect output is expressed with a “✗” symbol and an unknown result is shown with a “?” symbol.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Conditions</th>
<th>Expected Class</th>
<th>Model output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1, B1, C1, D1</td>
<td>1</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>A2, B1, C1, D1</td>
<td>1 or 2</td>
<td>?</td>
<td>✓</td>
</tr>
<tr>
<td>A3, B1, C1, D1</td>
<td>5,6,7 or 8</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>A1, B2, C1, D1</td>
<td>2</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>A1, B3, C1, D1</td>
<td>5,6,7 or 8</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>A1, B1, C2, D1</td>
<td>class 3</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>A1, B1, C3, D1</td>
<td>5,6,7 or 8</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>A1, B1, C1, D2</td>
<td>3,4 or 5</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>A2, B3, C3, D1</td>
<td>7 or 8</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>A2, B3, C1, D1</td>
<td>5 or 6</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>A2, B1, C3, D1</td>
<td>7 or 8</td>
<td>✗</td>
<td>?</td>
</tr>
<tr>
<td>A3, B1, C1, D1</td>
<td>5,6,7 or 8</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>A3, B3, C1, D1</td>
<td>5,6,7 or 8</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>A1, B3, C3, D1</td>
<td>7 or 8</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

Table 4 Results of testing from the CARE spreadsheet and EDM model

As can be seen in table 5, the EDM model of the CARE spreadsheet outperforms the CARE spreadsheet in the abnormal input test. The CARE EDM model classifies 9 of the 14 examples correctly comparatively the CARE spreadsheet classifies 3 of the 14 examples correctly.

The EDM model is therefore better at dealing with unusual input than the equivalent CARE spreadsheet. The reason that the EDM CARE model outperforms the CARE spreadsheet is because of the machine learning algorithm (neural networks) used to create the EDM model.
Neural networks possess the property ‘gradual degradation’ which means in circumstances where the network is being pushed to breaking point, failure is gradual rather than abrupt.

Since the CARE spreadsheet is based on Excel programming, failure is more abrupt than the gradual degradation observed in neural networks.

3.3 Advantages and disadvantages of using the example-giving technique

There are several key advantages and disadvantages associated with example-giving, this section aims to explore these issues.

3.3.1 Advantages

The main advantages gained from this approach are

- No spreadsheet programming required (cuts out mechanical and logic error)
- Removes human factor aspects such as Base Error Rate (BER) from the creation of the spreadsheet model.
- Example-giving is easier and more accurate than spreadsheet modelling
- EDM (example-giving using neural nets) is noise tolerant, i.e. user error
- The process of thinking up examples forces the user to consider the whole problem and subsequently may be the only method of reducing omission error.

Since the user is not required to do any programming, errors associated with programming the spreadsheet are effectively cut out. This is also true for Human Factor issues such as BER, since the user does not have the opportunity to commit BER in the creation of the spreadsheet model, BER is mitigated from the spreadsheet.

During the example-giving feasibility experiment described earlier in this paper, participants in the treatment (example-giving) and control (spreadsheet modelling) were asked how difficult they felt each task was. The results show that the treatment group found the tasks consistently easier than the control group. In fact the treatment groups results indicated that the difficulty of the tasks did not increase, conversely the control group show a consistent and progressive increase in difficulty. It is a reasonable assumption therefore that the treatment, example-giving is easier than spreadsheet modelling.

The comparison in accuracy show that example-giving is significantly more accurate than spreadsheet modelling, that is the participants were significantly more accurate at providing correct examples than creating correct spreadsheet models.

The use of Neural Networks as a means of implementing example-giving provides a useful advantage for EDM. Neural networks are “noise tolerant” which means that even when some of the data are incorrect or missing, the neural network is still able to operate with an acceptable level of accuracy. Experimentation contained in the authors PhD thesis [16] shows that EDM can tolerate a dataset with 15% noise, i.e. a data set that contains up to 15% erroneous or missing examples. Clearly this is useful benefit of EDM since BER may be present in the example-giving process.

One of the main benefits comes from the user generating the examples. The user is required to think up examples for the problem they wish to model in terms of attribute classifications. Attribute classifications are the input variables (attributes) and the resulting output (classifications). This requires the user to think in general high level terms about the problem. In similar approaches such as Test Driven Development (TDD) users write tests that resemble EDM input closely, the main difference being that TDD cases are written to test the software whereas examples in EDM are written to reflect the whole problem. This can result in a truer reflection of the problem that is not skewed by considerations for the software, hence this method may help reduce omission error.

3.3.2 Disadvantages

- Unfamiliar to spreadsheet modellers
- BER in example generation
- Neural nets black-box syndrome
- Rubbish in, rubbish out
- Variable quantities of examples required

Perhaps the biggest challenge to EDM as a methodology is the unfamiliarity of the technique to spreadsheet modellers. EDM is significantly different to spreadsheet modelling and this may hinder the useful application of the technique.
It cannot be ignored that the possibility of BER in the generation of examples is considerable, studies show that even with simple and repetitive tasks BER is a serious problem [17]. However, consider the ability of EDM to operate with noise, the impact of BER may be mitigated since EDM can still generate accurate models when up to 15% of the examples are erroneous.

Neural networks are often described as ‘black-box’ systems, i.e. although a system may operate correctly it is impossible to pick apart the workings internally. In the case of EDM this may mean that whilst attribute classifications are successfully turned into working models, it is impossible to backward engineer the process. This is deemed as a necessary cost to using neural networks in EDM.

Whilst EDM can significantly improve the accuracy of a problem, it is entirely dependant on valid representative examples being provided by the user. If the user provides examples that are an incorrect representation of the problem, the EDM model will reflect those incorrect aspects.

The number of examples required from the user depends on the problem wishing to be modelled. If the problem is relatively simple, such as the student grading system described earlier in this paper, experimentation from the authors PhD thesis [16] shows that as little as 25 examples are needed. However, with greater complexity more examples are needed, this relationship is difficult to quantify. It is clear when using this method if the number of examples is sufficient, if an inadequate number of examples is provided the resulting model will produce inaccurate results.

### 3.3.3 Areas of application

Although EDM performs well in the CARE algorithm model above, it is not suited to some types of spreadsheet model.

Spreadsheet models that are only numerical cannot be modelled using EDM, i.e. a spreadsheet that consists solely of mathematical calculations cannot be modelled using EDM.

EDM works well in spreadsheet models that make use of logic and mathematical calculation. In spreadsheets where logic is not present, EDM does not work. Spreadsheet models that use mathematics and logic are often ‘decision support systems’ that recommend some course of action based upon a number of numerical inputs and logical operations.

The CARE spreadsheet is a prime example of the type of spreadsheet EDM is suited to. The CARE spreadsheet combines some mathematical values with logic operators to give an assessment of patient mortality, morbidity and prolonged length of stay risk. These risks are bound to broad categories which the CARE spreadsheet indicates are true if the priori conditions are such.

### 4.0 Conclusion

The evidence contained in this paper demonstrates that example-giving as an alternative means to model decision support spreadsheets is feasible and more accurate than traditional spreadsheet modelling.

However, the example-giving technique (EDM) has very specific areas of application and does not work well in purely mathematical problems such as profit and loss accounts or balance sheets. Where the problem is based upon logic, EDM can be applied – the CARE spreadsheet is a good example of this. These types of spreadsheets are defined as decision support spreadsheets.

A full scale trial would further address the practical usefulness of the example-giving technique.
References