Empirical Modelling as an Experimental Problem Solving Application

0814909

Abstract

In this paper we evaluate the benefits of using Empirical Modelling (EM) as a framework for problem solving and the study of problem solving methods. The result of this project was the creation of a number of small models that demonstrate the capabilities of EM in educating users in general rules of problem solving. Each of these models tackled a different problem whose solution involves the use of these guidelines. We also examine the JS-EDEN environment with regard to problem construction and flexibility of solutions. We find that, whilst it has limitations, the JS-EDEN environment and EM are intuitive paradigms for examining problem via models and in which to teach problem solving techniques.

1 Introduction

In this paper we examine Empirical Modelling as a paradigm for problem solving by contrasting EM's benefits to our traditional problem solving methods. We examine EM as an extension to traditional problem solving through a number of models. Each model invites the user to solve a different problem posited by either Pólya (1990) or Mason et al. (2010). Each model also describes some of the problem solving guidelines suggested by these authors and applies them in an EM context.

The overall goal of this study is to evaluate both these problem solving techniques and the beneficial nature of problem solving using EM. We show that using EM can be an intuitive method to both problem solving and educating users about problem solving methods.

With regards to our initial proposal for this study a small change has been made.

This study no longer compares problem solving methods between different fields. It was realised that a true problem solving method should not be constrained to a single field of expertise. All the techniques described by our study can be applied as readily to Computer Science as Mathematics or any other field.

In general the puzzles that we examine are of a logical nature and reasonably simple. None of them rely on previous experience with their respective topics and all can be solved within a relatively short amount of time. However our models rely on the user becoming involved with the problem solving mechanics and we invite all users to approach the experience with gusto.

The structure of the remaining paper is as follows.

In section 2 we describe the background to our study and list the problem solving guidelines chosen for this study.

Section 3 describes the models created for this study. In this section we describe each model and state which problem solving guidelines each model examines.

Section 4 presents an analysis and evaluation of our models. We discuss whether they accomplish their purpose and reach conclusions as to EMs usefulness in problem solving and the teaching of problem solving methods.

Section 5 summarises and concludes our paper.

2 Background

Our primary inspiration and motivation for this study comes from the works of Pólya (1990) and Mason et al. (2010). Whilst both of these works approach problem solving from a mathematical point of view they describe general problem solving techniques that can be applied to any area of study. These guidelines or guidelines are generic such that they can be used within any field and yet remain helpful.

Pólya (1990) describes "Modern Heuristic" as the basis of problem solving. Modern heuristic is termed by Pólya as the study of problem solving, or more precisely, the study of the "process of solving problems". Pólya describes that such a process can be learnt and therefore improve the problem solving capabilities of any person, student or otherwise.

Mason et al. (2010) is based on the work of Pólya and also takes the view of education towards problem solving - that problem solving ability (or "Mathematical Thinking" as they term it) is improved with practise, structure and proper review.

Since Pólya and Mason both describe a great number of guidelines the study their work is beyond the scope of this study. As such we have chosen a subset of these guidelines to examine and analyse within EM. The guidelines we have chosen are as follows.

Analogy (Pólya, 1990) Recognition and use of similar or equatable facts or problems.

Auxiliary Problems and Elements (Pólya, 1990) Additional problems or objects that aid use in our understanding of the problem.

Specialisation (Pólya, 1990), (Mason et al., 2010) Using examples to understand a problem more clearly and to look for underlying patterns within the question.

Generalisation (Pólya, 1990), (Mason et al., 2010) The generalisation of patterns or rules that we have discovered to prove their truth and extend answers to related questions or fields.

Rubric (Mason et al., 2010) A set of keywords that help to structure our problem solving process.

Working Backwards (Pólya, 1990) The technique of starting from the desired solution and working towards the question's scenario.

The goal of this study is to look at how these guidelines can be used within EM and to evaluate how easily these problem solving methods can be described and taught within an EM program.

3 Model Creation

For this study we created three different models. Each of these models describes a problem to be solved using both the guidelines chosen in section 2 and EM methodologies. In this section we describe both our environment choice and the models.

Each model can be used or experienced in any order. For the purposes of this paper, however, they will be considered in alphabetical order.

3.1 JS-EDEN

There were a number of choices for the environment in which to implement our models. Each software environment was considered but it was JS-EDEN which was chosen overall. Our reasons for using JS-EDEN are, primarily,

- The ease of rendering polygons and images in order to make models more engaging.
- The ease of access through web browsing software.
- The additional input functionality in the form of sliders, buttons, combo-boxes etc.

Due to these reasons we believe that JS-EDEN is a more accessible environment for modelling, not only to the modeller, but for anyone experiencing the construal.

3.2 "Arithmagons"

The Arithmagons problem, as described by Mason et al. (2010), is based on a mathematical rule describing the relationship between and edge numbers of a triangle. The rule is given to the problem solver and the task is to determine another rule such that the corner numbers can be found with only the edge numbers shown.

This model takes the user through the stages of constructing the model and invites them to adapt it themselves. Whilst the model gives a number of examples and a solution to this puzzle, the user is also encouraged to try other examples and make their own observations. A screenshot of one state of the Arithmagon model is shown in figure 1.

The problem solving guidelines that this model encompasses are *Specialisation* and *Generalisation* - the cornerstones of problem solving. Specialisation is experimentation with examples such that relationships and patterns in the question can be discovered. Generalisation is the process of taking these conjectured patterns or singular examples and adapting them for a general case. In this example different sets of corner and edge numbers are our specialisations and the rule we seek to find is our generalisation.

The key observables in this model are the corner and edge numbers. It is the dependencies between these observables which are of most interest to the user in this problem. However it is up to the user to give agency to the problem, changing target edge and corner numbers and adding functionality where appropriate to aid pattern recognition.

3.3 "Fred and Frank"

Fred and Frank is a problem in Mason et al. (2010) which, in contrast to Arithmagons, has no mathemat-

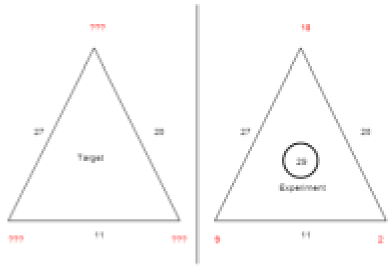

Figure 1: Screenshot of the Arithmagons Model

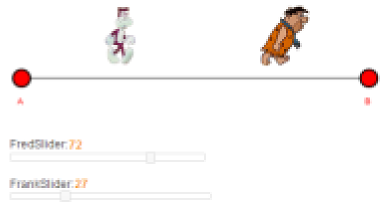

Figure 2: Screenshot of the Fred and Frank Model

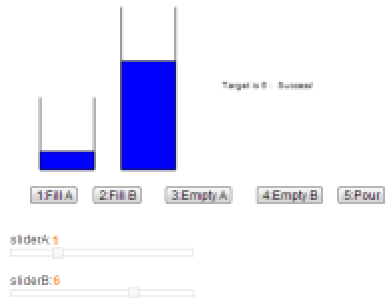

Figure 3: Screenshot of the Extended JUGS Model

ical background. The puzzle is a simple logical reasoning exercise, which may confuse at first, but is typically solved in a short period of time. The problem revolves around two racers with different rules of when they can walk or run.

In this model we look at *Analogy*, *Auxiliary Problems and Elements* and introduce Mason's *Rubric* structure of problem solving. Whilst simple the model and its accompanying notes seeks to show that even in the simplest of puzzles there may be a number of subtle steps that are taking place, perhaps subconsciously, by the problem solver.

This model is interesting in that the user can either extend it to model the race between the runners exactly, or it can be used as inspiration and guidance for the logical reasoning of the puzzle. In this way the model can either be the experimental device used for problem solving or a springboard for the imagination of the user.

A screenshot of this model is shown in figure 2.

The observables in this model are the runners and the slider inputs that control them. There are no real dependencies within this model. The important dependencies that exist are between the runners and the current logical reasoning state of the user.

By this we mean that, whilst the model itself does not wholly represent the puzzle that we are investigating, as the mental image of the problem of the user is changed the information represented by the runners also changes. The model has been created abstractly such that the user can interpret it either in terms of time, distance, speed or any other viewpoint that they are attacking the question from. In this way the key agent in the model is the user themselves.

3.4 "Working Backwards in JUGS"

This model is an extension of the JUGS model that was developed for JS-EDEN by Beynon (2012). Our extension adds to the original model by suggesting a new way of using the model for solving the jugs puzzle. This particular instance of the jugs puzzle is from Pólya (1990). A screenshot of the extended model is shown in figure 3.

The problem solving guideline that this extension examines is *Working Backwards*. We show that through the addition of new features or viewpoints we can use EM as a intuitive framework for working backwards from a goal to the start of a question.

As described in Beynon (2007) the jugs in the model are "passive agents" that represent the dependencies between contents, capacity and the various actions the user can perform. The various action buttons are agents as well changing and update the values of observables in response to the user.

To these we add more input mechanisms to exactly control the observable values the jugs represent. We enable the user to experiment more thoroughly with the model in order to discover an easier way of finding a solution to the puzzle.

4 Analysis

4.1 Ease of Implementation

Initially we felt restricted in our choice of problems on which to build a model due to the implementation of definitive scripts in EM.

For example we initially wished to explore the problem of "Kathy's Coins" from Mason et al. (2010). We wished to create a model such that the number of coins displayed could be varied by the user. Unfortunately, without object orientation, we were unable to find an implementable model that could be used. Instead we were forced to choose a simpler questions to examine.

However, overall, we had little difficulty creating and expanding our models from initial designs. The additional input devices that feature in JS-EDEN were of great use to us during our design. We also enjoyed using images within our models to make them more vibrant and engaging.

4.2 Model Utility

In general we feel that the models teach the guidelines posited by Mason and Pólya well. In particular we found that those guidelines that were coupled most closely to the field of EM were more easily described using models. These guidelines were typically those that could take advantage of dependency management and the flexibility of EM.

Traditionally multiple specialisations must be labouredly written out and formalised. In contrast Arithmagons EM's dependencies allowed many specialisations to be set up quickly thus allowing the user to gain a lot of knowledge from only small changes to observables. This also applies to generalisation as well. The users are able to test patterns that they have seen in their attempted examples against the model by rewriting observable values as dependencies. In this manner the user is able to incrementally build up a set of dependencies that get closer and closer to mirroring the puzzle or phenomena they are attempting to understand.

Fred and Frank, we believe, described the introduction of auxiliary problems and elements well - if only in the accompanying slides. In general it is simple to introduce new problems and features into a model. It is also helpful to be able to save the state of models using scripts at various stages of construction to allow us to better understand our work at a later date.

We also introduced auxiliary problems and elements in Arithmagons and Working Backwards in Jugs. However in Fred and Frank this is done at a more abstract level, bouncing new problems and thoughts off of the model instead of adapting and extending the model itself. This describes how EM can not only be used to investigate phenomena through a virtual model, but also be a starting point from which to jump to new logical conclusions, without the need of additional features.

In general the Rubric steps and the idea that many subtle steps are often needed within the simplest of questions seems to be well taught to the user. Typically the user of the model will have worked out a solution to the puzzle very shortly after reading the question. However by going through the puzzle at a steady rate, as the Rubric dictates, the user can see what processes their mind may have subconsciously gone through during their reasoning. This allows the user to better reflect on their thought process during subsequent problem solving.

Finally Working Backwards in Jugs provided users, whether familiar with the previous model or not, with a different viewpoint on the problem stated in the initial model. In our extension we showed that adding a supplementary features to a model can give users different routes of problem solving to experiment with.

We also were able to show how affording the user more flexibility typically results in the discovery of more efficient and helpful ways of solving and approaching problems.

4.3 EM and Problem Solving

The freedom of exploration and experimentation that can be taken from EM seems to be helpful for problem solving. As shown in our models some problems are most easily solved by adding in new rules, elements and questions into our problem solving process. It is simple to pull these additions into a model due to EMs flexibility for redefinition at any point in time. In this way models based on EM, rather than traditional software engineering, will usually represent the user's experience with a problem better and at a lower cost to the modeller.

In general problem solving revolves around the discovery of the underlying relationship between aspects of a question. This perfectly mirrors EM's basic components of Observables, Dependencies and Agency. In particular its dependency management system is of greatest use as it can be used to slowly move from examples to general rules that describe the phenomena occurring in the question. This system allows users to be creative and try a number of solutions until they are satisfied that they have found the correct result.

The reasons why EM is helpful in encouraging creative or free-form thought is due to ability of the software to provide the modeller with as much freedom as possible. The tools that can be used in an EM environment allow the modeller to mould a construal into any state desired. This cannot be accomplished with traditional programming as they require that as much of the environment is automated away from the user. In doing so traditional computing trades interaction for efficiency.

In contrast EM seeks to allow humans as much freedom to fulfil their needs and different viewpoints. It does this by providing an environment where there is no intended absolute viewpoint for the model. This means that EM does not pretend to predict how users might use, adapt and extend the model.

This is comparable to problem solving as it, as a field, embraces that everyone has different quirks, experience, knowledge and logical processes.

By enhancing our problem solving with EM, rather than other traditional tools, we are not limiting our problem solving abilities by the design of the initial model creator. As everything can be changed and extended in a model the user has the capacity to decide, or learn, how they personally attack problems. They can decide for themselves which of the guidelines is most natural to them and, in the case of a problem that grabs them, continue to expand and extend the model with their experience of the question.

Therefore, due to its flexibility, freedom and dependency management, EM is an intuitive and helpful device for both problem solving and the teaching of problem solving methodologies.

5 Conclusions

Over the course of our study we have found EM to be a helpful addition to traditional problem solving methods. The greatest benefit of modelling is the flexibility it affords to the user and the freedom that the user has to extend and change models.

Models also form good representations of the user's current mental progression over the course of a problem solving exercise and can be used both as simulation devices and more abstract devices to inspire users.

Whilst it was not possible to model all the problems we desired in this study, due to limitations within definitive scripts, we are satisfied that models can be created such that they can teach users general problem solving techniques within a free-form setting.

We conclude that EM is an intuitive paradigm in which to both practise and learn problem solving.

References

W. M. Beynon. Computing technology for learning-in need of a radical new conception. *Educational Technology and Society*, 10(1):94–106, 2007.

W.M. Beynon. Jugs in JS-EDEN. [online] Available at: `http://www2.warwick.ac.uk/fac/sci/dcs/research/em/teaching/cs405/jugsinj-se/` [Accessed 26 January 2013], 2012.

J. Mason, L. Burton, and K. Stacey. *Thinking Mathematically*. Pearson Education Limited, 2nd edition, 2010.

G. Pólya. *How to solve it: The classical introduction to mathematical problem-solving*. New York: Penguin Books, 2nd edition, 1990.