Is Empirical Modelling Puzzling?

0208873

Abstract

Su Doku, ”The Rubik’s cube of the 21st century” is the faster growing puzzle phenomenon in the world, replacing traditional crosswords, alphawords and lateral thinking games in nearly all major newspapers, Smith (2005). A craze that is almost impossible to ignore, with clubs, worldwide tournaments, books, software and even television programmes dedicated to the puzzle. In this paper we introduce the concept of the Doku-games, a family of puzzles that share many properties with Su Doku, and outline a script of definitions to create a comprehensive Doku-game model.

1 Introduction

This paper illistrates the benefits of using Empirical Modelling techniques to develop and design puzzles and games for education use, with the specific example of the recent Su Doku craze. We develop a model that can generalize the Su Doku format to create a family of games which we shall call Duko-games. This generalization will enable us to alter the size and rules of the game, enabling any m x n grid size, as well as specialize the model to any of the existing Duko games currently in use. It will also provide a single tool to enable the verification of whether the grid of symbols has been correctly completed.

1.1 Concept of the Doku-games

We will refer to the catagory of games studied as Doku-games. Examples of such games are:

- Su Doku (3 by 3)
- Shi Doku (4 x 4)
- Roku Doku (6 x 6)
- Go Doku (5 x 4)
- Killer
- Samurai
- Latin Squares

Each of these puzzles is played by a single player on a m by n grid. The grid is split into what are called regions represented by the number in the brackets. Within each region the player must place every symbol belonging to the alphabet as not to break the constraints of the game. The object of the puzzles are to place every alphabet symbol into each region, column and row such that they appear only once. The catagory of games are called Doku games for this very reason, Doku meaning singular in Japenese. Conventionally the symbols in the alphabet are represented by numbers, but can just as easily be letters, colours or any other symbols that can be easily distinguished from each other. For simplicity we shall use numbers in all examples.

2 Analysing the Doku-games

In order to develop the Doku-games environment it is important to observe and analyse actions in human play. Doku-games are played by a single state changing agent, with a single view of that state. Changes of the state are made via interaction with the agent. The player is concerned by two interactive processes. The first process is the physical constraints of the player, for example where they are able to place a symbol. The second, which involves a higher level of observation, is whether the changes to state they are about to make are valid changes, and will not result in the breaking of constaints established by the rules of the game.

There are 2 common strategies employed by the agent when making such high level iteraction with the environment.

The first of which is to *choose a square on the grid and discover which numbers can be placed in that square*.

Figure 1 shows the squares in which a player must observe in order to make such a move. They must consult the row that the square is in, the column, and the region in which the square falls.

In this example it can be seen from consulting the

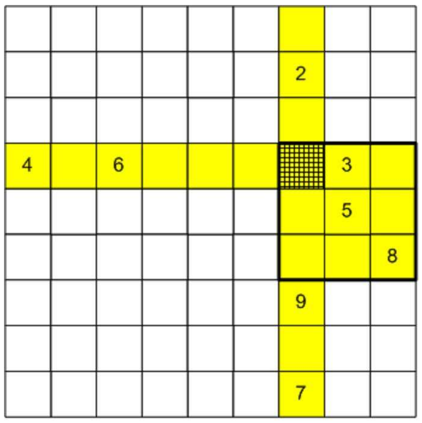

Figure 1: Strategy One

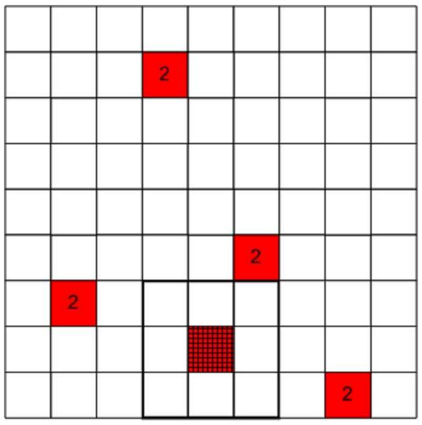

Figure 2: Strategy Two

rows that the marked square cannot be a 4, 6 or 3. Likewise for the column, the marked square cannot be 2, 9 or 7. Finally from consulting the region in which the marked square is contained 3, 5 and 8 are also not allowed. Taking the union of these 3 sets gives us a conclusive set of values that cannot be present. In this example this set is {2,3,4,5,6,7,8,9}. Now removing these elements from the list of alphabet symbols {1,2,3,4,5,6,7,8,9} (as this is a Su Doku game), we are left with a single symbol represented by the number 1.

Strategy one has been implemented in the model accompanying this paper.

The second strategy involves *picking a specific symbol and trying to determine where it could be placed*.

Figure 2 shows how such a placement might be derived. In this example the symbol 2 has been chosen, and the player has picked the region in the center at the bottom of the grid. Observing that there are 2's in all the regions in the same column and row as the selected region, the player is able to determine where to place the symbol in the selected region.

How to do Sudoku, Vorderman (2005), which explains a number of these strategies, is the best-selling book in the country, but certain things about the game cannot be explained using formal language. It is our experience in doing the activity that helps us progress further, and that formal language has no definitive meaning, a view stongly suggested and argued by James (1911). It is with interactions between the agent and the environment that aids learning greatest.

In terms of the Doku-games this experience will help us complete more difficult puzzles and decrease the time it take for us complete them. A formal description of how to play the puzzles with different strategies may help us learn, but it cannot progress us past a certain stage.

It is also important to understand the processes that human agents undergo when interacting with such puzzles. This understanding may help us develop automated agents that could be used to demonstrate how to complete a puzzle or provide hints to players who are struggling. Standard automated Su Doku solvers and helpers tend to simply reveal a symbol on the grid, rather than interpret the players methods to provide suggestion. The automated agents may also be able to provide answer to those questions raised in section 4.2.

3 Context

The perception and analysis of games using Empirical Modelling tools has been adressed before in the modelling of a family of games resembling noughts-and-crosses, W.M. Beynon (1994). This paper demostrated the creation and benefits of establishing a platform in order to extend and experiment with OXO games. This paper extends these principles by showing how they can still apply in a different family of applications, extending work on other Su Doku, and grid models previously created, Harfield (2003), K. King (2005). Previous scripts to model Doku-games have only concerned Su Doku, and although these have been very useful to consult we have not reused any of the definitions found within.

4 The Doku-game Model

The following section will document the current functionality of the Doku-game model, how this has been achieved through dependancy, and how these dependancies can be changed to experiment with the environment.

4.1 Current Functionality

Using the EDEN, SCOUT and DoNaLD notations the Doku-game model presents an interactive user interface that allows the user to create and verify a Doku grid of any length alphabet. They are able to alter the occurences of each alphabet symbol in each row or column. The user is also provided with a mechanism for suggesting a valid number for each square, implementing strategy one, and mechanisms to show whether errors are present in either the column, row or region and whether the grid is complete.

4.2 Dependancy

The unusual property of the Doku-games is that every alphabet symbol is dependant on every other. In a good puzzle, there is only a single solution, and the placement of alphabet symbols can only happen in a distinct order. Although the Doku-game model does not constrain where the users are allowed to place the alphabet symbols, so the player can make mistakes, the placement of incorrect symbols will not allow the user to complete the puzzle. The following list is a list of all the dependancies in the Doku-game model provided.

- **Grid size** - The size of the grid is dynamically changed via alteration of three key variables. These are:
  alphabet_length - The alphabet_length variable determines the basic dimensions of the grid. The grids x and y length (grid_size_x and grid_size_y respectively) are directly linked to the value of the alphabet_length.
  occurences_row - For example if the value of the row occurences is altered from 1 to 2, the value of the grids x length is doubled and the grid redrawn on the change of this value.
  occurences_column - The same applies for the column occurences.

- **Allowed symbols** - The symbols allowed in the grid are dependant on the alphabet_length variable. Players are contrained to only enter those values which are in the alphabet.

- **Suggestion mechanism to implement strategy one** - The model takes the union of three lists corresponding to the current row, column and region. These lists are dynamically changed depending upon where the user has the mouse placed. The union of these lists is then subtracted from the alphabet_list to make the suggestion.

- **Completion criteria** - The completed observable is dependant upon the values of the three completion criteria, those of no symbol repeated in each distinct row, column or region, and the placing of the correct number of characters in the grid. The results of each of the these criteria can be observed through the interactive environment. These results take the value of false if an error is present in the particular section that they represnet, and true when no errors are present. In turn, each of these criteria is dependant on the values placed in the grid structure and their corresponding lists representing which numbers should be placed in each one.

4.3 Experimental Platform

Much like the OXO models mentioned earlier, Beynon (2005), this Doku-game model should not be seen as a program for playing Doku-games, but as a platform/laboratory to conceive and develop new variants. However through extension of the model and its dependancies the user could ask and discover answers to such questions as:

- *What happens if we remove lines or squares in the grid?* - The dependancies governing the completion criteria could be altered so that when a marked square or line were encountered, the rules do not apply.

- *What would happen if I added extra contraints to the placement of numbers?* - The dependancies concerning the criteria for correct columns, rows, and regions could be altered. New dependancies could even be introduced that disallowed diagonal placements. These could then be intergrated with the completion mechanism.

There are also a number of unanswered questions on the mathematics of sudoku as posed by Gupta (2005).

- *What is the smallest number of clues possible in order to make a satisfactory solution?* - A satisfactory solution is one where the player can deduce the whole grid with no guess work. Currently for 6x6 grids it is believed to be 9, and for standard 9x9 Su Doku it is believed to be 17. It is also believed that for any super Doku MxM, where M is greater than 11, the minimum number of clues required is greater than M. The answer to this question does not require extra development of the model. However adding dependancies to implement the other strategies of play and encompassing them into the suggestion meachanism, would greatly simplify this task.

- *Is there a Sudoku for all tilings of the grid for a prime alphabet length?* - When you have a prime alphabet length, it is not possible to create a rectangle or square to contain the alphabet symbols, so the regions are redrawn into what are known as polyominos. Two examples of tilings these polyonimos into a 5x5 grid are shown in Figure 3.

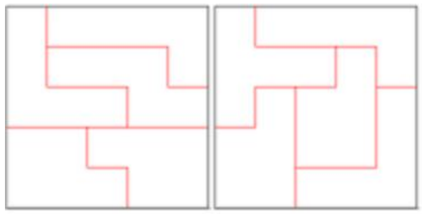

Figure 3: Tilings for 5x5 grids

Through extension of this model, the user could expand the grid data type to include a reference to the region as well as the symbol currently placed in that square. The region correctness dependancy could then be altered to encompass these new references allowing the user to answer the above question.

- *What is the smallest number of clues required to create a satisfactory sollution in a prime grid?* - As was mentioned above for a 6x6 grid, it is believed that 9 clues are required. However on the following 5x5 grid, with the polyominos tilings, the minimum number is 4.

The user can extend the model to answer this question by adding the new grid data type as mentioned above. They can experiment with different tilings of polyominos on different prime number grids, and through this experiment it may also be possible to disprove the conjecture mentioned previously.

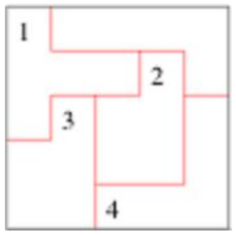

Figure 4: Satisfactory Solution for 5x5

5 Benefits of Doku-games

The huge appeal of Doku-games is that they do not require extensive general knowledge or strategies, so can be played by anyone. However because of this arguments have been made that the family of games we refer to as Doku-games do not involve any worthwhile teaching elements. Arguments have also been strongly expressed contrary to this view:

- **Mathematics** - At the very basic level, Su doku helps children recognise and remove numbers from a list of possibilities, a basic set operation. It teaches the concepts of rows and column, a feature that you would not expect children as young as 6 to be concious of. Variants of the classic Doku models such as Killer, involve addition and subtraction to identify numbers for each square in the grid.

- **Brain Activity** - The process undertaken by anyone trying to complete such a puzzle can help increase brainpower and consentration. Analysing possible moves and their consequences also teaches logic and reasoning to children at a very early age. Teachers who assign Doku puzzles to their students have also noticed increase in confidence amoungst shy and nervous children, as the puzzle is able to give children a common interest encouraging communication. The benefits of playing such puzzles are not solely associated with children. Performing such brain exercises has been linked to reduced memory decline in the elderly and is even claimed by some to help prevent the onset of brain disorders such as Alzheimer's (Black, 2005; Wheldon, 2005).

Arguably more important than any of the above reasons, is that creating and completing such puzzles is fun. They introduce people to the concepts of recreational mathematics and logic. What is technology supposed to be if not to keep us entertained?

6 Evaluation of EM Tools

This section will look at the advantages of using the Empirical Modelling tools to develop puzzle applications. It will also comment on a fundamental problematic issue concerning the development of scripts and the impact this problem may have on the methodology.

6.1 Benefits of using EM Tools

Introducing software versions of the puzzles combat many of the problems associated with puzzles on printed media. Puzzles can be automatically generated, allowing a seemingly infinite number of games. They can be verified by comparing the input with the stored values from the generation of the puzzle, a task previously done manually by the user. Using the computer enables the player to mark and remove symbols easily, allowing mistakes and experimentation with strategies, features previously not well supported. There are even mechanisms for providing hints to players who are stuck, and highlighting incorrect moves. These features are what every Su Doku fan needs to learn the game, experiment and play puzzles to their hearts content.

The Empirical Modelling tools provide a huge number of benefits to the software engineer compared to other programming methodologies. As well as being able to implement the above features, the tools are also able to provide the following benefits:

Experimentation with the environment. Just as the Su Doku craze was created from the game Latin Numbers, the tools enable the user to change the definitions and rules of games to create variants that will challenge different mental processes. Current commercial programming languages do not allow such flexability. Empirical Modelling tools allow a kind of evolution to the program, to grow alongside the users experience and creativity. It provides the ability to create an infinite number of 'updates' that can be easily added or removed to suit the user.

Customisability. Properites can be altered adhoc, so it is possible for any user to select the level of difficulty appropriate. In the classroom, teachers could assign smaller grids to those children less experienced instead of excluding them as most of the puzzles current might. Most current implementations of such games do not allow this customizable nature.

Code reuse. The structures developed in the model could be easily lifted and used in another application. For example the dynamic creation of the grid could become a standard for producing puzzles based on grid-like structures. This feature of the tools would reduce development time.

6.2 Areas of Improvement

Although the EM tools allow the developer to use some very powerful ideas, we would like to raise a number of concerns that we believe are holding back development.

Without a simple method of removing definitions and dependancies, true experimentation is not easily conducted. For example in the current Doku-game model the user may wish to remove a completeness constraint. However it would be extremely difficult to remove any of the constraints governing the correctness of columns rows and regions with the tools currently available. It also proves difficult to re-initialise lists, a function vital when resetting the board to play another puzzle.

One final point is that of scalability. It is easy to see how divergent programs could become if they were given to developers with a very small base of definitions. However as more definitions are added to this base script, does it make it harder to truely experiment with the environments? As more definitions are added the new developer is bounded by the application the original developer hand in mind. Coupled with the issue raised above, this would mean producing truely divergent models would not be easy.

7 Conclusion

In this paper we have discussed the benefits of using the Empirical Modelling tools as a foundation for developing variants of popular puzzles using the recent Su Doku craze as an example. We have also discussed an area in which these tools need to be improved in order to gain weight in the software development discussions.

Such crazes as Su Doku are a perfect portal to get the Empirical Modelling methodology recognised. Getting Empirical Modelling into the classroom is a perfect way to introduce future software developers to the benfits of such methods. If there was ever such a time to do this, it would be now.

Acknowledgements

I would like to thank Meurig Beynon and Steve Russ for their tuition in the thoeries and practices of Empirical Modelling. I would also like to thank Anthony Harfield, Karl King and George Efstathiou for use of their models, and all those who taught, and were on-hand to answer questions about, the Empirical Modelling tools during the module.

References

M. Beynon. Empirical modelling for the single agent. 2005.

E. Black. Eat eggs and play sudoku if you want to be brainier. World Wide Web: http://news.scotsman.com, 2005.

S. Gupta. Mathematics of sudoku. World Wide Web: http:theory.tifs.res.in/ sgupta/sudoku, 2005.

A. Harfield. grid.e, 2003.

W. James. Some problems of philosophy, 1911.

G.Efstathiou K. King. sudoku.e, sudoku_builder.e, possibles.e, 2005.

D. Smith. So you thoughtsudoku came from the land of the rising sun... World Wide Web: http://observer.gaurdian.co.uk, 2005.

C. Vorderman. *How To Do Sudoku*. Ebury Press, 2005.

J. Wheldon. How sudoku keeps the mind yound and healthy. World Wide Web: http://www.dailymail.co.uk, 2005.

M.S. Joy W.M. Beynon. Computer programming for noughts-and-crosses: New frontiers. 1994.