Educational Technology: Complex Data Structures Through Construals

1008010

Abstract

This paper investigates the use of Empirical Modelling techniques in educational technology for teaching complex data structures. The result of the project is an exploration of using Empirical Modelling to teach how red black trees function showing the capabilities of this approach for this domain. A comparison with previous work and alternative approaches is discussed. We find that despite the approach necessitating some restrictions in what can be achieved, the overall approach produces a successful and innovative construal, which encourages further research into this area.

1 Introduction

In this paper we explore the use of Empirical Modelling in the education of data structures, specifically self balancing trees. First a brief discussion on the relationship between Empirical Modelling and education in Section 1, followed by a description of the specific interest of this paper, red black trees. The construal which has been constructed to demonstrate the application of Empirical Modelling techniques to this area is discussed and explained in Section 2. The next section discusses the extent to which the constual accurately models the data structure and the limitations and extensions which are known. Finally the initial proposition of the paper is examined in light of the construal and its effectiveness.

1.1 Empirical Modelling

Empirical Modelling provides an alternative way of thinking than the traditional methods of programming and interaction with computers. It centres around the construction of construals; artifacts of the modeler's understanding of the subject domain. These contruals are comprised of observables, dependencies between the observables and agency to interact with the construal. This paper is accompanied by a construal model of a red-black self balancing tree written in Eden, a definitive notation language based on Empirical Modelling principals.

1.2 Educational Technology

One application of this way of thinking is within the educational domain. As construals are constructed on the basis of dependencies with associated agency, they are highly interactive. This allows for the user to explore the subject domain without any restrictions, any of the dependencies can be altered and the impact of these changes seen immediately reflected in the observables. Furthermore, the process of constructing a construal also provides the modeler with the opportunity to explore the domain, in which it resides in a alternative fashion, allowing for their personal knowledge of the subject domain to be explored and enhanced also.

There has already been various discussions regarding the application of Empirical Modelling techniques to the educational sector as well as the development of construals to demonstrate its application.[1][2][3] This paper adds to this growing work as well as builds upon previous ideas and construals and aims to show the possibility of teaching complex ideas in computer science through the use of construals.

1.3 Data Structures

An important part of an education in computer science is a grounding in the theory of not only simple data structures but more complex and useful ones.[4] However, grasping these concepts is regarded as a difficult task, especially when confronted with more convoluted data structures which have complex logic behind them.[5] Personal experience has been consistent with this, especially when studying the advanced data structures and algorithms modules in previous years.

One would postulate from experience that a significant reason for this is the reliance on rigid textbook proofs as the basis for understanding. This level of detachment from the use of the theory could possibly be a significant reason for these perceived difficulties. The research done at Virginia Tech support this theory, "lecture instructors typically draw on a board, trying to illustrate dynamic processes... many students have a hard time understanding these explanations"[5]. Therefore, removing this barrier to the comprehension of these concepts could be found through the use of construals for practice based learning to accompany the standard teaching methods employed. This practice based learning approach is favored in educational research.[6]

2 Red Black Tree Construal

2.1 Red Black Trees

The need to store and access data efficiently is the reason behind complex data structures.[4] This leads to the use of progressively more complex structures with more specific contexts but which provide improvements in performance. Binary search trees are one of these well studied stat structures and is commonly taught when introducing the key concepts of data storage and retrieval.[4] However, a major drawback of simple binary search trees is the tree becoming unbalanced, one side of the tree becoming significantly longer than the other reducing the efficiency of searches.

Red black trees are an extension of binary search trees and part of the self balancing binary search tree family. As well as the usual binary search tree insertion rules, extra conditions and properties for each node are used to re-organise the tree each time a new node is added.[7] These constraints assure that the tree is evenly balanced to keep the lowest possible access times, $\mathrm{O}\big(log\ n\big)$ in practice.[7]

Each node in the tree is assigned a colour, red or black, hence the name of the data structure. There are three main conditions which need to be imposed to ensure the tree is balanced:

- Every node is red or black
- The children of a red node are both black
- Every path from a node to a leaf has the same number of black nodes[7]

The third constraint is the main reason that the tree remains balanced and this is achieved by various re-colourings and rotations of the nodes each time a new node is inserted. There are 5 main cases which are identified as requiring various actions to maintain the third property.[7] These are however symmetrical cases and so result in more cases handled when implemented.

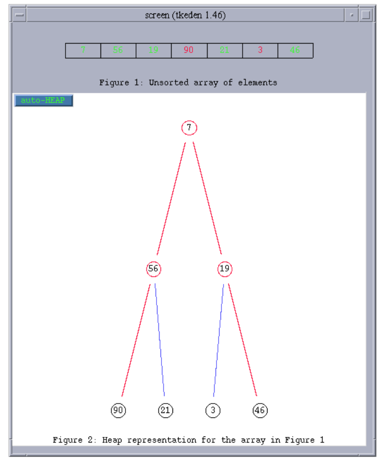

Figure 1: The heapsort construal[8]

2.2 Heapsort Construal

The inspiration for choosing this domain is a previously constructed construal based around the heap concept. The heapsort construal displays an initially unsorted heap with the heap conditions expressed through the colour of the edges between nodes. The construal also is able to visually automate the sorting process through the user clicking to update the observables as the heap is sorted.

There is much similarity between the heapsort and the red black tree, they both aim to sort data into readily accessible structures and furthermore on a visual level are strikingly similar. This is one aspect which the construal exercise was hoped to bring out, the relationship between two differing data structures being identified through the construal in unexpected ways. Similarly the relationship between the array and the tree, in both construals, can be seen through the dependency between the order of the values in the heapsort construal and the order in which items are added to the tree in the red black tree.

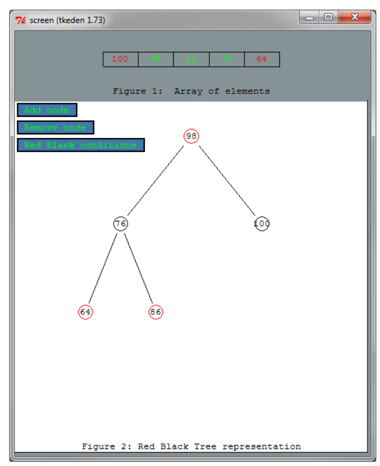

Figure 2: Red black tree construal in Eden

2.3 Observables

All of the nodes and edges in the tree and the values in the array, are observables in the red black tree construal. Their values are updated as the user interacts with the model and the changes to their states can be seen immediately. This feedback allows for the exploration of the relationship between the observables and how the theory of the data structure effects the components which it is comprised from.

2.4 Dependency

The colours of the edges between each node are dependencies which are based on the binary tree insertion conditions similar to the heap conditions in the heapsort construal. Red indicates that one or more of the conditions is not met in the current position and so the node is moved to different positions until a correct position is found.

This dependency allows for the interaction with the rules which govern binary tree insertion in a interesting way. The colour of the lines make the type of rule violation clear, by either being a single red edge long, indicating that the node is on the wrong side of the parent, or multiple red edges indicating that the node should be on the complete other side of the tree.

2.5 Agency

To interact with the construal there are buttons on the left side of the window which allow for the addition and removal of nodes. This is the main way in which the model is manipulated, once added the process of finding the correct place within in tree is shown by the user clicking on the model. The final button applies the red black tree rules to the newly added node, showing how the nodes are re-coloured and rotated.

It is also possible to interact with the construal through the Eden interpreter, here definitions can be altered, values in the tree changed and observables added. This allows for extra scope for interaction than would traditionally be possible and no restriction on what can be done to the model. By changing certain observables the effects upon the binary tree and red black tree rules can be investigated allowing for a more inquisitive exploration of their logic.

3 Evaluation

3.1 Advantages

The construal successfully displays the process of creating and maintaining a red black tree within an Empirical Modelling framework. The resulting trees keep to the required conditions and also display the process of initially adding nodes to a binary search tree. Hence, without having the initial intention to achieve this, the construal is educational for understanding how binary search trees function, an unexpected yet welcome addition to the construal.

As previously mentioned the ability to manipulate the observables and dependencies through the Eden interpreter is a significant advantage to using an Empirical Modelling framework and provides the initially intended flexibility for the user to explore the model in their personal way, building up their own understanding. Finally, the automation of both the binary tree insertion and the application of the red black tree reorganisation is an achievement. Building on the ideas used in the automation of the heapsort construal this successfully shows the user the whole process of maintaining the tree whilst also providing control through agency in the buttons and clicking to advance the process.

Whilst developing the construal, the understanding of the domain, and the relationship between the rules of red black trees and the reasoning behind them, became much more clear. Furthermore, the relationship between a basic binary search tree and the more complex red black tree also became better understood. From this experience of developing the construal, one can extrapolate that a similar benefit would result in interacting with it to learn about red black trees. Hence, to some extent the use of Empirical Modelling principals to education within complex data structures has been successful.

3.2 Limitations

There are various compromises which were accepted when using the Empirical Modelling methodology in creating this construal with the currently available tools. For example, as the observables are defined in the model, dynamically changing the size of the tree is problematic. In cases where adding nodes to the tree in a binary search fashion required more than a depth of 4 in the tree, the construal would not be able to accommodate this. Therefore, the number of items which could be added to the tree was limited from the original 7 in the heapsort construal down to 5 to avoid this situation.

Furthermore, the rotation of nodes to maintain the red black properties occurs immediately. A more desirable approach would be to move them gradually to emphasise the reasons behind the way in which nodes are re-arranged. This proved to difficult to include in the construal due to way in which the observables are defined.

3.3 Extensions

A desirable extension to the construal would be to enable larger sets of data to be added to the tree through dynamic generation of observables when the current tree is not large enough to display the required nodes. Although the dependencies and logic are already in place for this. There is also scope to add the ability to view which red black tree conditions are being rectified with each rotation or re-colouring within the construal. This may aid in the comprehension of the individual rules which are enforced.

4 Conclusion

The success of the model in conceptually delivering a complex construal in an accessible way leads to desiring more development of such work. Empirical studies with students learning these concepts for the first time would be a welcome next step to determine how much of an impact this sort of resource can make. Perhaps extending the construal to encapsulate more data structures to show the interdependency between them and the common similarities such as those seen within the heapsort and red black tree construals would be useful in unifying this with previous research.

The project set out to explore the use of an Empirical Modelling methodology to the teaching of complex data structures and has achieved this aim. Overall the success of the construal is currently ambiguous due to the lack of empirical studies, however, the personal experience of the writer and understanding gained from the modelling process, seems to suggest that this is a profitable approach for learning and hence more research into the area is recommended.

References

[1] R. Boyatt, A. Harfield & W. M. Beynon, Learning about and through Empirical Modelling, Department of Computer Science, University Of Warwick (2005)

[2] W. M. Beynon, Empirical Modelling for Educational Technology, Department of Computer Science, University Of Warwick (1997)

[3] EM for Learning - A Modeler's Experience, WEB-EM 07, Department of Computer Science, University of Warwick (2011)

[4] K. Mehlhorn, P. Sanders, Preface, Algorithms and Data Structures - The Basic Tool Box, Springer (2008)

[5] S. Hall, C. A. Shaffer, Evaluating Online Tutorials for Data Structures and Algorithms Courses, Virginia Polytechnic Institute and State University (2013)

[6] R. Alexandre, A framework for practice-based learning applied to computer programming, Department of Computer Science, Faculty of Sciences, of University of Porto (2012)

[7] A. Mantler, Constructing Red-Black Tree Shapes, Department of Computer Science, University Of North Carolina (2001)

[8] W. M. Beynon, J. Rungrattanaubol & J. Sinclair, Formal specification from an observation-oriented perspective, Department of Computer Science, University Of Warwick (2008)