Empirical Modelling for constructionist learning in a Thai secondary school mathematics class

Antony Harfield
Naresuan University
Phitsanulok, Thailand
antonyh@nu.ac.th

Meurig Beynon
University of Warwick
Coventry, UK
wmb@dcs.warwick.ac.uk

Abstract— **Constructionism advocates that learning takes place when the learner is actively engaged in building or 'constructing'. Empirical Modelling provides a web-based environment that enables learners to build software artefacts in an unusually open-ended and flexible manner. In this paper, we explore the potential for using the principles and tools of Empirical Modelling to support constructionist learning in a Thai secondary school context. An activity was implemented for a Mathayom 3 mathematics class, on the topic of linear inequality that enables students to explore a simple model of a number line. The activity encourages students to go beyond the typical "question/answer" exercise and attempts to give students the capability to build a number line model themselves within the Empirical Modelling environment. The paper discusses the benefits and limitations of the implemented approach.**

Keywords— **web-based education, interactive environments, secondary school, mathematics education**

I. INTRODUCTION

Empirical Modelling (EM) is an approach to building software artefacts (often referred to as *models*) in an environment that is unusually open-ended and experiential [1]. An EM model consists of a set of definitions that are conceptually similar to the dependencies between cells in a spreadsheet. The user works on a model though a process of continual refinement and elaboration of the definitions. As demonstrated by previous work, EM is highly applicable as a constructionist learning environment [2], where the learner engages in the exploration or construction of a model. Examples from existing projects have shown how EM can be applied to learning in school [2,4], in university [3] and more generally to lifelong learning [5].

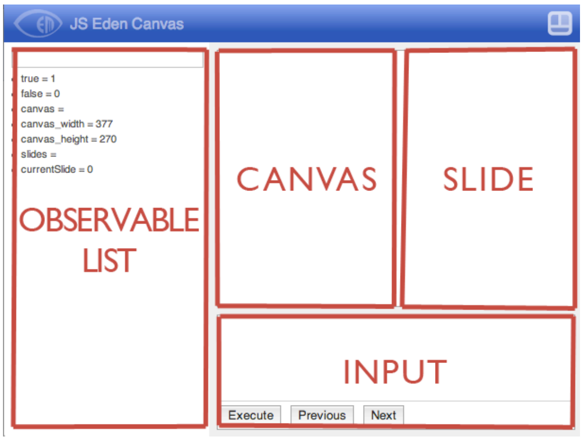
Since the first constructionist learning environments introduced by Papert and others in the 1980s, a wide range of methods, tools and educational topics have been explored, from early logo turtle geometry through to more recently topics in the sciences and the arts. Despite constructionism often being described simply as 'learning by making', Papert promoted "a sense of constructionism much richer and more multifaceted" [6]. Empirical Modelling as an approach to constructionist learning is well-suited to provide support for the many rich and varied experiences that a learner typically engages in.

To motivate the need to elaborate 'learning by making', consider a young child who encounters brick-based toys (e.g Lego) for the first time. It is unlikely the child will begin to construct a house with the bricks immediately. Initially the interest might be in connecting the bricks together, and such interest might need to be provoked by a parent or elder sibling who demonstrates how the bricks connect. Hence, instruction may precede construction. Often there is repetition of a basic skill before it becomes familiar, and then later the child can use it with confidence to engage in a more open-ended task such as building a house. The key observation is that there are ‘auxiliary’ activities (instructing, repeating, exploring) that complement the ‘making’ activity. Together, these activities could form the basis of constructionism in a broader sense, as alluded to by Papert. The idea that making as an activity is insufficient is well recognized by Rau et al. [7] whose work demonstrates that ‘fluency’ plays an essential role.

In this paper, the authors describe how Empirical Modelling can provide support for some of the auxiliary activities that presuppose construction as well as the making itself. An example of using EM for teaching mathematics in a Thai secondary school illustrates the need for instruction, repetition, and exploration within a context of construction.

II. A WEB-BASED ENVIRONMENT FOR EMPIRICAL MODELLING

The EM tool described in this work is called JS-Eden and was initially developed by Tim Monks [9]. It builds on a history of tools based on the EDEN notation developed by researchers at University of Warwick. EDEN (Evaluator of DEfinitive Notations) is a notation for describing sets of definitions using the EM concepts of observables, dependency and agency [1]. JS-Eden is a Javascript and HTML 5 web application that offers similar functionality to the desktop EDEN tool. It runs in modern web browsers (standard version available at http://jseden.dcs.warwick.ac.uk). The version of JS-Eden described in this paper was modified by the authors for use in schools (by removing unused features and simplifying the user interface).



- true = 1
- false = 0
- canvas =
- canvas_width = 377
- canvas_height = 270
- slides =
- currentSlide = 0

Fig. 1. Simplified user interface layout of JS-Eden

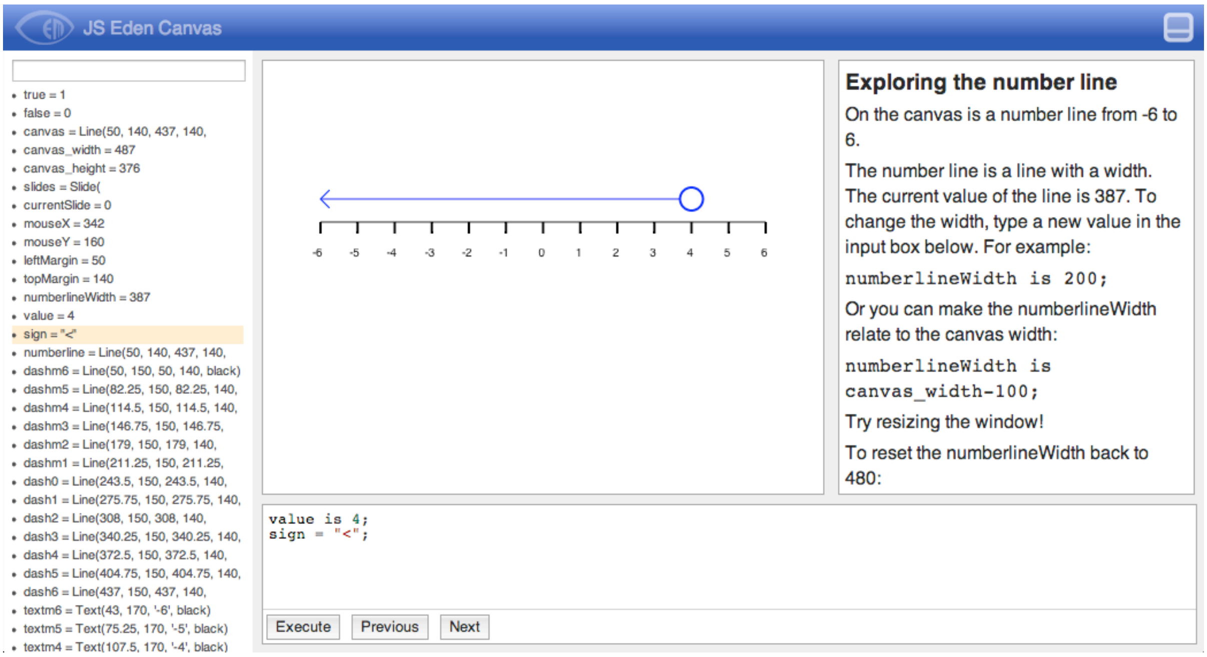
The user interface for JS-Eden comprises four main areas: the input box, the canvas, the slide deck and the observable list (as pictured in Figure 1). A typical mode of use for a learner is to: a) observe the visual appearance of the model on the canvas; and b) follow the instructions/examples/tasks shown on the slide. The learner interacts with the model by: a) using the mouse/keyboard on the canvas; b) modifying definitions on the observable list; or c) entering new definitions into the input box.

A JS-Eden session would usually begin by loading an existing model. The set of definitions for the model would set up the canvas and the instructions on the slide. From which point a learner could exercise the model, undertake some task or perhaps explore extending the model further.

III. THE NUMBER LINE MODEL

One of the topics in the Mathayom 3 curriculum in Thailand (14-15 year olds) is linear inequality. The content of this topic revolves around the understanding of a ‘number line’ which can be described as a horizontal line with numbers marked on at regular intervals (e.g. -10 to 10). A linear inequality such as ‘x > 2’ can be represented on the line as an open circle above the number 2 with an arrow pointing right (for ‘greater than’).

A number line model in JS-Eden can be expressed as a set of definitions describing the dimensions, position and appearance (e.g. line width and colour) of the number



- true = 1
- false = 0
- canvas = Line(50, 140, 437, 140,
- canvas_width = 487
- canvas_height = 376
- slides = Slide(
- currentSlide = 0
- mouseX = 342
- mouseY = 160
- leftMargin = 50
- topMargin = 140
- numberlineWidth = 387
- value = 4
- sign = "<"
- numberline = Line(50, 140, 437, 140,
- dashm6 = Line(50, 150, 50, 140, black)
- dashm5 = Line(82.25, 150, 82.25, 140,
- dashm4 = Line(114.5, 150, 114.5, 140,
- dashm3 = Line(146.75, 150, 146.75,
- dashm2 = Line(179, 150, 179, 140,
- dashm1 = Line(211.25, 150, 211.25,
- dash0 = Line(243.5, 150, 243.5, 140,
- dash1 = Line(275.75, 150, 275.75, 140,
- dash2 = Line(308, 150, 308, 140,
- dash3 = Line(340.25, 150, 340.25, 140,
- dash4 = Line(372.5, 150, 372.5, 140,
- dash5 = Line(404.75, 150, 404.75, 140,
- dash6 = Line(437, 150, 437, 140,
- textm6 = Text(43, 170, '-6', black)
- textm5 = Text(75.25, 170, '-5', black)
- textm4 = Text(107.5, 170, '-4', black)

Exploring the number line

On the canvas is a number line from -6 to 6.

The number line is a line with a width. The current value of the line is 387. To change the width, type a new value in the input box below. For example:

```
numberlineWidth is 200;
```

Or you can make the numberlineWidth relate to the canvas width:

```
numberlineWidth is
canvas_width-100;
```

Try resizing the window!

To reset the numberlineWidth back to 480:

```
value is 4;
sign = "<";
```

Fig. 2. The number line model

line shown on the canvas, as well as the meaning or interpretation of the number line to the modeller. For example, the basic line might be defined by the EDEN notation as:

```
numberLine is Line(20, 100,
    canvas_width-20, 100, “black”);
```

Here, ‘numberLine’ is defined as type Line with a startpoint (x,y), an endpoint (X,y) and a colour. Note that the x-coordinate X of the endpoint depends on the width of the canvas, so the number line maintains a suitable width whenever the canvas size changes. Furthermore the meaning of number line might also be defined in the model:

```
arrowDirection is (sign == ”>”) ?
                    “right” : “left”;
```

The ‘sign’ is the ‘greater than’ or ‘less than’ that represents the inequality. The definition for ‘arrowDirection’ does not necessarily have any impact on the appearance of the model, but is part of the underlying interpretation by the modeller. However, it is usual for the appearance to reflect the meaning of the model. In this case, there would most likely be a line representing the arrow that would be dependent on ‘arrowDirection’ in the same way that the width of the line is dependent on the width of the canvas.

IV. DISCUSSION

The number line model was introduced in 2 two-hour sessions to 33 students in Mathayom 3 from the English programme at Princess Chulabhorn College Phitsanulok. The students had little previous exposure to linear inequalities and the students had no prior experience of the JS-Eden tool. (They had some rudimentary knowledge of programming from an introductory programming class taught as part of a computing course.) The first session was dedicated solely to understanding the JS-Eden environment and becoming familiar with modifying and creating definitions. In the second session the number line model was the basis for a number of problems that the students were asked to solve. Informal feedback from the first session was somewhat negative with students failing to see the relevance of learning a new tool. However feedback from the second session was more positive, with a number of students showing a desire to continue using the tool in subsequent weeks. Several students went beyond the required set problems to develop their models further.

Three ways in which the number line model was used in the classroom are explained, each demonstrating an auxiliary activity that contributes to a broader sense of constructionism.

A. Establishing a context ("introducing")

In a school environment, instruction is likely to precede construction, as the learner needs a certain amount of support or scaffolding when starting a new topic. JS-Eden was used at the beginning of the session as a demonstration environment for the teacher to explain the concept of a number line (referred to as a 'guided tour'). One quality of the JS-Eden environment is support for a slide deck that can be used by the teacher to introduce the model. It provides an interactive substitute to a more traditional presentation environment. The slide deck can also be used by the learner, for example, to revisit the guided tour given by the teacher. On the right hand side of Figure 2, the slide is showing an explanation of the concept of the number line, and instructions on how to exercise the number line model. As the slide deck is part of the model, it is natural that the content of the slides can be changed depending on the state of the model. For example, a slide might highlight the current values of a selected definition, or a new slide might reveal further activities on completion of a particular step in the tour.

B. Consolidating understanding ("repeating")

Watching or following the guided tour of the number line model is unlikely to be sufficient to achieve competency in the topic of linear inequalities. The student must practice, and this often means repeating similar types of activity multiple times. One early example of JS-Eden supporting such repetition was in the first session when there were many tasks involving creating lines, as this was considered a crucial skill for building your own number line model. In contrast to the compile-run paradigm of most programming tools, JS-Eden is well-suited to making small changes to a model on-the-fly, building a model one definition at a time. The nature of the repetition is not necessarily copying the same action every time, but repeating a pattern of action making small adjustments each time. In the second session, the sign and the value of the inequality were changed regularly in order to exercise the student's knowledge of how sign, value and visualisation interact. Such repetition might be cumbersome in traditional tools, but in JS-Eden revising or refining definitions is the essential mode of interaction.

C. Probing the boundaries ("exploring")

Repetition takes a learner to the point where they can do something confidently without much thought. For a deeper understanding, the keen learner might engage in his or her own self-directed exploration. From the number line sessions at the school, some students appeared reluctant to go beyond the set tasks and needed some encouragement. In the second session, the students were given some relatively open-ended tasks such as using extra labels or lines to describe the current state of the number line. A small number of students were able to make a little progress in the short amount of time. For example, some students created colour coding for labels to show which direction the number line was pointing, while others added arrows or other visual artefacts to show the state of the game.

V. CONCLUDING REMARKS

One of the key qualities of JS-Eden as a learning environment is that the deeper exploratory activity naturally extends from introduction and repetition activities. The idea that learning progresses from primitive concrete interaction with an artefact (private experience) towards exploring a formal understanding (public knowledge) is proposed by the Experiential Framework for Learning (EFL) [8]. Although the types of activity suggested by the EFL are more numerous than the three categories described in the current work, they share a similar ambition for recognising a broader perspective on the use of technology for learning. EM provides a set of principles that are well aligned to supporting rich and varied types of learning activity described by the EFL. This paper not only demonstrates how JS-Eden can be used as a constructionist environment for teaching mathematics to secondary school children, but also highlights some of the qualities of the environment for different types of learning and teaching activity.

ACKNOWLEDGMENT

The JS-Eden tool was developed by Tim Monks and extended by members of the Empirical Modelling Research Group, University of Warwick, UK.

REFERENCES

[1] W.M. Beynon. Modelling with experience: construal and construction for software. Chapter 9 in Ways of Thinking, Ways of Seeing (ed. Chris Bissell and Chris Dillon), Automation, Collaboration, & E-Services Series 1, Springer-Verlag, January 2012, ISBN 978-3-642-25208-2, 197-228.

[2] W.M. Beynon and A. Harfield. Constructionism through Construal by Computer. Constructionism 2010, The American University of Paris, August 2010.

[3] A. Harfield, W.M. Beynon and R. Myers. Web Eden and Moodle: an Empirical Modelling approach to web-based education. In Proceedings of the Eighth IASTED International conference on Web-Based Education, 16-18th March 2009, 272-278.

[4] Ilkka Jormanainen, Meurig Beynon, Erkki Sutinen. Understanding open learning processes in a robotics class. Proceedings of the 9th Koli Calling International Conference on Computing Education Research (ed. Pears and Schulte), Technical Report 2010-027, Information Technology, Uppsala University, Sweden, November 2010, 51-54.

[5] W.M. Beynon and A. Harfield. Lifelong Learning, Empirical Modelling and the Promises of Constructivism. Journal of Computers, Volume 2, Issue 3, May 2007, 43-55.

[6] S. Papert and I. Harel. Situating Constructionism. Constructionism, Ablex Publishing Corporation: 193-206. 1991. URL: http://www.papert.org/articles/SituatingConstructionism.html.

[7] M. Rau, V. Aleven, N. Rummel and S. Rohrbach. Sense Making Alone Doesn't Do It: Fluency Matters Too! ITS Support for Robust Learning with Multiple Representations. In S. Cerri, W. Clancey, G. Papadourakis & K. Panourgia (Eds.), *Intelligent Tutoring Systems*. Vol. 7315, pp. 174-184. Berlin / Heidelberg: Springer. 2012.

[8] W.M. Beynon. Empirical Modelling for Educational Technology. Proc. Cognitive Technology '97, University of Aizu, Japan, IEEE, 54-68, 1997.

[9] T.R. Monks. A Definitive System for the Browser, MSc dissertation report, Computer Science, University of Warwick. Online at: http://www2.warwick.ac.uk/fac/sci/dcs/research/em/publications/mscprojects/timmonks/