Designing educational software: The case of grid algebra

This item was submitted to Loughborough University's Institutional Repository by the/an author.


Additional Information:

- The final publication is available at Springer via http://dx.doi.org/10.1007/s40751-016-0018-4.

Metadata Record: https://dspace.lboro.ac.uk/2134/21615

Version: Accepted for publication

Publisher: © Springer

Rights: This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) licence. Full details of this licence are available at: https://creativecommons.org/licenses/by-nc-nd/4.0/

Please cite the published version.
Designing educational software: the case of Grid Algebra

Dave Hewitt, Loughborough University

Abstract To be successful educational software needs to be carefully designed in order to provide constraints and freedoms (Greeno 1994) which relate to its educational aims, restrict or enable learners and teachers to carry out certain actions, and be based upon a clear set of beliefs related to the complex dynamic of teaching and learning. This paper seeks to show the contraints and freedoms built in to the design of Grid Algebra; software which offers an intuitive and novel approach to early algebraic learning, and how they have been informed by a clear set of beliefs and frameworks. The paper shows how the design has taken into consideration known misconceptions that many learners exhibit with early algebra work and how the three frameworks of arbitrary and necessary, subordination and embodied cognition have been fundamental to the design of both the software and some of the activities used with the software. The paper finishes off with considering limitations of the software as well as some strengths that its design offers, and poses questions to consider about the design of this and other resources.

Keywords Embodied cognition; Technology; Algebra; Misconceptions; Notation.

Introduction

This paper discusses some of the key ideas which lay behind the design of the software Grid Algebra¹. In doing so it offers an example of how design can tap into some deep rooted knowings that all learners have, be based upon ways in which fluency has been achieved within our everyday lives and, more specifically within the topic of algebra, how known misconceptions can be addressed or avoided. The software offers an original and intuitive approach to teaching algebra; a topic found by many teachers to be ‘hard to teach’. Following discussion of background issues, I gradually introduce some of the key design features of the software and how these relate to established difficulties learners experience when learning algebra. This is followed by articulation of three frameworks which underpin the whole design of the software. I finish by considering some of the limitations of the software as well as summarising the features which make this an intuitive approach to the early learning of algebra, and raise questions not only related to the design of this particular software but more general questions about the design of resources which could be based upon the frameworks used here.

Background

There has been a considerable amount of research in the decades towards the end of the 20th century which has indicated difficulties learners experience with algebra (for example Küchemann 1981; Booth 1984; Filloy and Rojano 1989; Herscovics and Linchevski 1994; MacGregor and Stacey 1997). The influential Cockcroft report in the UK (DES 1982, p. 60) indicated that “algebra is a source of considerable confusion and negative attitudes among pupils” and as teachers we want to make sure that “algebra is not a meaningless game with 26 letters” (Freudenthal 1973, p.290, his emphasis). The difficulties include how the equals sign is viewed (Kieran 1981), the meaning given to letters (Küchemann 1981), the reading of formal mathematical notation (Kirshner 1989) and the need to treat an algebraic expression as an object as well as a process (Booth 1984; Tall and Thomas 1991).

Studies have shown that working with letters in a meaningful way can be achieved by relatively young learners (Carraher et al. 2001) and so the issue arises whether adolescents’ difficulties with algebra may not lay solely with the learners themselves but with teaching approaches which impact negatively on their success with

algebra (Brizuela and Schliemann 2004). Some teaching practices are not helpful when making the shift from numerical work to algebraic activity. These include: the almost exclusive use of arithmetic statements which have only a single number on the right-hand side of the equals sign; the use of calculators where pressing an equals button on a calculator gets a single numerical result; and always being expecting learners to carry out calculations when faced with an arithmetic expression. McNeil and Alibali (2005) offer a change-resistant account where such entrenched operational patterns persist and contribute to learners’ difficulties with equations. Different images have been offered to try to assist learners with algebra. These include the use of cups and tiles (Caglayan and Olive 2010), an empty number line (Dickinson and Eade 2004), a spreadsheet (Ainley 1999) and a balance image (Vlassis 2002). Each of these images can offer some support for learners but none of these offers the opportunity for learners to work directly with standard mathematical notation. As a consequence there would need to be a separate learning process for learners to learn to write and read standard notation. There have been attempts to do this when using some of these images (Falcao 1995; Bills et al. 2006) but the journey to using standard notation remains a far from straightforward transition for learners. Although learners are quite capable of creating their own notation (Steffe and Olive 1996) the “inverse process of interpreting symbols to unfold mathematical concepts challenges all learners” (Sáenz-Ludlow and Walsguth 1998, p. 153). This is reflected in the fact that using standard notation is sometimes seen as the final part of generalising activities, for example finding a rule for a geometric pattern (Warren 2006). The sometimes unconventional notational form with which a learner writes a rule carries with it their own awareness and embodied experience of working on a problem (Radford 2002); as such the form of the notation, albeit non-standard, matters. So why would a learner feel it necessary to write it in a different way just because a teacher asks them to do so? In contrast, Dörfler (2006) suggests seeing notation as an example of an object, or diagram, through which we learn mathematics. As such notation is not an afterthought but an inherent part of mathematical activity.

The software Grid Algebra is designed to help learners become fluent with standard notation whilst developing their algebraic abilities towards, amongst other things, solving linear equations. Behind the design is a perspective of algebra as stressing properties of operations and attending to mathematical structure, with or without the inclusion of letters. Gattegno (1988, p. 77) describes algebra as essentially concerning “operations upon operations”. Within a school arithmetic context, this would involve a shift of attention from carrying out operations in order to do arithmetic, to attending to the operations themselves and operating upon those operations. For example, becoming aware of order, inverse and equivalence of operations can be used in general arithmetic situations such as multiplying out brackets or solving equations. The sense of generality and focus upon operations, rather than numbers, brings a sense that particular numbers are irrelevant and consequently letters can be introduced and used with relatively little conceptual difficulty for learners.

Grid Algebra calls upon mathematical notions such as order and inverse which are known intuitively through moving objects and making journeys in their daily lives. The software makes use of the idea of movement on a grid of numbers to create and work with expressions written in standard notation. Elsewhere (Hewitt 2014b) I have described the intricate relationship between the intuitive world of physical movements on the grid and the abstract world of mathematical symbolism with which the movements are associated. Learners can shift between one world and the other whilst developing their confidence in working with the mathematical notation. The software has enabled relative young learners (9-10 years old) to go from never having met the use of brackets in a mathematical context, a division line in notation instead of the division sign ÷, nor a letter in an algebraic context, to solving linear equations within just three lessons (Hewitt 2012). At the end of the third and final lesson, learners were offered a choice of one of two sheets to complete; the first had a series of equations to solve as in Fig. 1, and the second also had equations to solve but these were placed within the visual context of movements around the Grid Algebra grid (Fig. 2). More explanation of these movements will be given later. Fig. 1 gives the work of a pair of learners from this mixed ability class of 9-10 year olds which was completed at the end of the third lesson. One thing to note is that this sheet contained all their working and as such they were able, as were many others, to solve these equations without any intermediate steps.
One effect of using the software was that learners become very familiar with standard notation and were used to reading order within the notation and also identifying inverse operations in the appropriate order as well. Even the weakest learner in the class was able to write solutions on paper to equations involving up to four operations where the paper challenge included a visual picture of the Grid Algebra grid (Fig. 2).

The design of the software is centred around two main aspects: (a) it tries to address some of the standard misconceptions related to learning algebra; (b) it is rooted within strong theoretical frameworks which inform the nature of the feedback learners receive, how learners gain fluency with standard notation within a relatively short amount of time, and how it accesses and formalises intuitive notions which are deeply embedded within all learners. At this point I will say something briefly about each of the three theoretical frameworks but will leave more detailed discussion until the reader has already become aware of some of the functionings of the software, which will be done through discussing some standard misconceptions learners have with algebra.

The first framework is arbitrary and necessary (Hewitt 1999, 2001a, 2001b) which divides the mathematics curriculum respectively into those things where there is choice and which have been socially agreed (names and conventions), and those things which can be argued that they must be how they are (properties and relationships). It is argued elsewhere (Hewitt 1999) that learners need to be informed of what is arbitrary and Grid Algebra does this by providing the way in which mathematical expressions are written in formal notation. Also learners do not need to be informed of what is necessary and so Grid Algebra does not provide any answers or carry out any calculations. Instead activities are offered from which learners can become aware of mathematical properties and relationships.

The second framework is that of subordination (Hewitt 1996). This concerns the way in which learners have to use the formal notation provided by the software despite their attention being focused elsewhere and even though they may not be confident with interpreting that notation. The way in which the software provides neutral but interpretative feedback (Edwards 1998) allows the learner to revaluate their interpretation of the notation and develop confidence and fluency with that notation through its subordinated use to the given activities.

The third framework is embodied cognition (Varela et al. 1993). This concerns the claim that mind and body are not separate and that many of the ways in which we think about things have roots within bodily experiences. Grid Algebra accesses deep experiences all learners have experienced through movement. Through making a
strong association between, for example, the notions of order and inverse (a) as they are experienced within movement, and (b) as they can be expressed within mathematical activity and notation, learners’ new understanding of mathematical situations is supported through their already established, albeit implicit, knowing within the context of movement.

More detail about each of these frameworks will be discussed and examples of how they have informed the design of Grid Algebra will be given later in the paper. This is because the depth of how they have informed the design is best seen once the reader is already aware of the key functionality built into the software. The functionality will be developed gradually through not only a description of the grid itself and movements within the grid, but also through how the design has taken into consideration some of the standard misconceptions that many learners experience when learning algebra. This paper does not have the space to address every design feature; however, key ones are discussed here. The structure of this paper addresses these design features whilst also gradually introducing the reader to the software, its functionality and some of the key mathematical activities with which learners engage whilst using the software. It is important to realise that the design is not just restricted to the functionality of the software but also in some of the activities envisaged in conjunction with the software use. The functionality of the software provides certain affordances and constraints (Greeno 1994) and as such allows certain activities to take place. However, envisaged activities also inform the design of the functionality. So, activities and functionality are interlinked. Indeed Grid Algebra has many computer generated activities built into the software, although only a few are mentioned here. There is also an Interactive mode where a teacher or learner can create their own explorations on the grid. Before starting on the two aspects mentioned above, I will address the design of the grid and its related symbolism, both of which form the basis of all subsequent activity. In the initial discussion of design issues, the examples I offer relate to numbers and it is later on that I bring in letters. There a number of reasons why I have done this. Firstly, it gives a sense of the gradual shift in attention, from numbers to operations and letters, for learners as they engage in activities. The second reason relates to the way in which I view algebra as being concerned with “operations upon operations” (Gattegno 1988, p. 77). As I discuss design issues, there is a key moment when attention shifts from numbers on the grid to operations which establish relationships between numbers on the grid. Thus, the design of the software is such that the focus is on operations rather than numbers per se.

The structure of the grid

The static grid

School algebra is based upon generalisation of the four arithmetic operations; addition, subtraction, multiplication and division. Fig. 3 offers part of the first six multiplication tables, with the numbers in the left-hand column indicating which tables they are.
On starting up the software, the grid is empty but there is a *Numbers* button within a toolbox which, when pressed, will fill up the grid with numbers. The default is that it will fill up the grid as in Fig. 3. Slider bars can be seen and these can be used to view other multiplication tables if scrolled down, or further numbers within each table if scrolled to the right. The figure shows six rows but a smaller number of rows can be on view if that is desired, right down to a single row. Not all numbers have to be on view at any one time. There is a *Rubber* which can rub out any particular number or collection of numbers. It is also possible to start with a blank grid and choose a number from a *Number Box* and drag that number into a cell. The software is designed so that the number is only accepted into a cell if it is in the times table for that particular row. So, in an otherwise blank grid, the number 12 can be placed anywhere in rows 1, 2, 3, 4, 6 or 12. If it is placed in a different row, then it appears in a light grey colour with a *No Entry* sign in the cell. This allows someone time to consider why it is not being accepted and then on the next click of the mouse the number will drift off into a *Bin* which appears on the screen. When the grid is empty, any number can be dragged from the *Number Box* into any cell in an appropriate row. There is a choice of allowing just positive integers only in the *Number Box* or to allow negatives as well. With an empty grid, the value of each cell is undetermined and so an icon of an empty grid appears on the top-left of the screen to indicate this. When a number, such as 12 is placed in a cell the grid is then defined and there is no longer any choice as to which number should go into any other cell. To indicate this, the grid icon becomes full of numbers. If the *Numbers* button on the toolbox were pressed now, it will fill the grid with numbers based upon the fact that 12 is in that particular cell. Thus, if 12 were in row 4 (the four times table), then the number to the left of it would be 8 and the number to the right would be 16. The number above it would be the equivalent number in the three times table, 9, and below it would be the equivalent number in the five times table, 15.

There are many activities which can be done at this stage to help learners become familiar with the structure of the grid. For example, a grid full of numbers can have one number rubbed out and a learner has to drag the appropriate number into that blank cell. Or the grid can be blank except for a number in one cell. Someone can Highlight another cell on the grid (colours are available from the toolbox) and the task is for a learner to decide what number must be in that highlighted cell and drag the number from the *Number Box* into that cell. Such
activities are significant as the explicit ‘outer’ meaning (Tahta 1981) is about placing the correct number into a specific cell, but the implicit ‘inner’ meaning concerns shifting a learner’s attention away from numbers and on to arithmetic relationships between the cells. In order to carry out such tasks, a learner begins to work out arithmetic operations which link the position of the given number to the position of the highlighted cell. This is a pre-cursor for what is to follow.

Movement on the grid

School algebra places an emphasis upon arithmetic operations rather than numbers themselves. For example, ‘multiplying out brackets’ (the distributive law) is a property of operations, not of numbers. Thus, to work towards algebraic activity requires attention to be placed on operations. Initially, the grid tends to stress numbers as it is these which are seen. However, operations exist in the relationships between the numbers in the grid. For example, if I were to look at the number 1 and consider what I can do to 1 to get to 2, I can either think of this as 1+1 or 1×2. The phrase ‘to get to’ can also be interpreted visually as a movement from the number 1 in the grid to get to the number 2 in the grid. This gives a sense of making a journey on the grid with operations being the journeys taken from one number on the grid to another number on the grid\(^2\). The two options of 1+1 and 1×2 can both be accommodated since multiplication tables can be seen both horizontally and vertically. Thus a decision was taken to view the horizontal as additive and the vertical as multiplicative. This is an arbitrary decision but what is important is that once the decision is taken, it remains consistent in the design of the software. So operations concern both the arithmetic relationship between numbers and the physical movement on the grid from one cell to another. The software allows both of these perspectives to occur simultaneously through a person being able to click on the number 1 cell and physically drag this across either horizontally to the number 2 cell on the right, or vertically to the number 2 cell below (see Fig. 4). The first movement will reveal the expression 1+1 in the cell which had the number 2 in it, whilst the second movement will produce the expression 1×2 in the cell below which also had the number 2 inside it. In this way, the software changes the static grid with a focus on numbers into an interactive dynamic grid with the focus on movement and operations.

\(^2\) In fact it is really about one cell on the grid to another cell. The awareness that it is about cells rather than particular numbers brings a sense of generality which allows letters to be introduced in an intuitive way. This is discussed later.
Movement to the right is shown consistently as addition, no matter which row. The particular row is significant however, since moving one cell to the right in row 4 will result in adding four and not just one since it is in the four times table. The inverse operation to addition is subtraction and this is matched with the inverse movement of moving to the left rather than the right. This means that learners’ intuitive sense of inverse within movement can assist learning of inverse within arithmetic operations.

Downwards movement is consistently interpreted as multiplication. When starting from the one times table, moving downwards to any other times table is straightforward. For example, moving down from 5 in the one times table to 30 in the six times table would result in \(5 \times 6\) being shown (see Fig. 4). Consideration needs to be given to other vertical movements which do not start in the one times table. For example, starting with the 12 in the three times table and dragging it down to the number 24 will show \(12 \times 2\). This has involved a downwards movement of three rows to produce \(x^2\) whereas it would have been only a downwards movement of one row if it has started in row 1 (the one times table). There is a significant pedagogical issue here to help learners become familiar with multiplication on the grid. Knowing which multiplication table a number is in is important and so this is one reason why there is an additional left-hand column added on the grid to label the multiplication table for each row.

A key design issue was deciding what would happen if someone wanted to drag a number from the three times table to the four times table. This would mean multiplying by \(\frac{4}{3}\) and other possible multiplications could involve decimal numbers with a greater number of recurring digits. An alternative would be representing such a multiplication as \(\times \frac{4}{3}\). This would avoid the messiness of having numbers with several decimal places and stay more in touch with the particular rows involved. However, there would be other issues involved such as whether such fractions get simplified or not. Clearly, the example above of moving from 12 to 24 is arguably more naturally represented as multiplying by two and not by \(\frac{6}{3}\). So, should other movements, such as moving from...
the four to the six times table be represented as $\times \frac{3}{2}$ rather than $\times \frac{6}{4}$? There is no ‘correct’ answer to such designs issues and the final decision was based upon pedagogical considerations rather than purely mathematical considerations. I saw this software being used in primary as well as secondary schools and with learners with a wide range of attainment. I also wanted attention to be on generalities concerning operations and as such I did not want learners to be distracted by trying to work out why certain non-integer numbers appeared. Consequently, I decided that a downwards movement which resulted in multiplication by a non-integer would not produce an expression at all. Such movements result in an ‘X’ appearing temporarily. This means that all expressions produced on the grid are made up of operations with integers. It has the added advantage of allowing a learner to see that if they start with the two times table then as they drag downwards it is only every second row that they get an expression appearing (see Fig. 5).

As with addition and subtraction, division is produced by carrying out inverse movements to those for multiplication. Thus division is an upward movement and is represented symbolically with the division line rather than the symbol $\div$ (see Fig. 4 for an example). The decision for this is that the software heads towards work with expressions using letters and as such the division line notation is important for what is to follow.

The introduction of movement is key to shifting attention from numbers to operations. With the focus on operations, the shift into algebraic activity begins.

**Difficulties with algebra**

I will consider four well researched difficulties which learners experience when beginning to learn algebra: process/object; introduction of letters; parsing expressions; and the equals sign. For each of these I will discuss briefly some issues from literature before relating those to the design of the software.

**Process/Object**

It is common for younger learners to be asked to carry out arithmetic tasks where the emphasis is on developing the fluency to be able to add, subtract, multiply and divide. The emphasis is on carrying out a calculation correctly and as such a single numerical answer is the desired response. Learners come to expect answers to be a single entity and that if something is stated which involves an operation then this must be a question, not an answer. It is no surprise then that learners have difficulty seeing an expression such as $5 + n$ as an answer. Instead it is viewed as a question and as a consequence a learner might attempt to do something to it in order to make a single entity; for example by conjoining to produce $5n$ (Booth 1984; MacGregor and Stacey 1997). Falle (2007) argued that learners need to be exposed to arithmetic expressions in different ‘unclosed’ forms where operation signs are still seen visually within the expression. She also felt there should be a greater
emphasis on the structure and meaning of expressions. Sfard (1991) identified that an expression such as 
\(3(x + 2) - 1\) needs to be viewed as an object in its own right as well as a process to be carried out and Gray and Tall (1994) talked of the ability to view expressions in this flexible way as proceptual thinking.

I have already mentioned that a number can be dragged horizontally or vertically to create a new expression. That new expression can then be dragged once again so that another expression is formed of two operations (since it has been dragged twice). This can continue so that expressions of some complexity can be created. Fig. 6 gives an example of the number 36 being dragged several times to produce the expression

\[3\left(\frac{36+12}{2} - 4\right) + 18\]

\[\frac{6}{-2}\]. Each time one of the expressions is dragged, it is being treated as an object, one that can be clicked on and dragged somewhere. This is different to how an expression might be treated if it were written on paper. The form of the expression serves a key purpose; it is a historical artefact which reveals the journey undertaken. Any simplification of this expression, through calculation or transformation, would destroy that information. So the form of the expression is of upmost significance. Sfard and Linchevski (1994, p. 211) said that an "equation requires suspension of actual calculations for the sake of static description of relationships between quantities." Here, Grid Algebra stresses the relationships in terms of the operations carried out on the number 36 and it is these operations which are the focus rather than any desire for calculation.

![Fig 6](image.png)

**Fig 6** The number 36 taken on a longer journey

A key activity within Grid Algebra is to try to re-create such a journey when all expressions are rubbed off except for the original number, in this case 36, and the final expression, 

\[3\left(\frac{36+12}{2} - 4\right) + 18\]

\[\frac{6}{-2}\]. With such an activity, the only clue for a learner is the form of the final expression and as such this becomes something which is examined carefully and has particular significance. So, unlike so many usual class activities involving
expressions with operations, there is a specific need not to carry out any of the calculations but to see this as an object of significance in its own right. Further, as Falle (2007) recommended, the activity helps focus on the structure and meaning of the expression.

Koedinger et al. (2008, p. 366) comment that “Abstract representations, such as algebraic symbols, are concise and easy to manipulate but are distanced from any physical referents”. The physical referents within Grid Algebra can be seen both as movements on the grid and mathematical operations (Hewitt 2014b). The symbolic expression appears in the very place where the associated movement is made. The symbolism and referent are closely related spatially. Thus there is a strong sense of physically acting upon the symbolic expression itself in creating expressions through physically picking up an expression and moving it.

The combination of seeing an expression as an object, as mentioned above, along with other activities such as a learner being asked to drag a single number from the Number Box into the cell with \( \frac{36 + 12 - 4}{2} + 18 - 2 \), allows learners to treat such an expression as a process as well as an object and so helps to develop what Tall (1991) describes as a proceptual way of thinking.

**Introduction of letters**

I have observed many a lesson where the learners appear quite confident with the work until a letter is introduced. Then there are questions about what does that letter mean and why is it there. In a Year 8 (12-13 year olds) class there were two examples written on the board as follows:

**Example 1:**
\[
x + 3 = 9
\]
\[
x = 6
\]

**Example 2:**
\[
x + 7 = 10
\]
\[
x = 3
\]

When I asked one of the learners whether she understood what was written she replied: “I don’t understand, I just copy it down”. I then asked what the ‘ex’ was on the board, to which she replied “What ‘ex’? That’s a times.” Interpreting letters when they are first met is problematic for many learners (Dickson 1989). Letters can be interpreted in many different ways, sometimes being associated with an object rather than representing a number, whether unknown or variable (Booth 1984). Indeed, one of six categories of letter usage identified by Küchemann (1981) was representing an object. MacGregor and Stacey (1997) saw examples of letters being interpreted as an alphabetical code, where \( a = 1, b = 2 \), etc., where values of expressions were based upon the order value of the letter within the alphabet. So the introduction of a letter can be problematic and my own experience has been that if attention is placed upon the letter when it is introduced this invariably leads to learners questioning what it means and why it is being introduced. As a consequence, in considering activities with Grid Algebra, I wanted attention to be elsewhere when a letter is introduced. Such an activity I will now describe.

I have already mentioned a key activity of trying to re-create a given expression on the grid. In addition I have also mentioned that a key feature of Grid Algebra is shifting attention from numbers onto operations. This means that within the re-creating activity, attention is on which movements/operations are needed and the order of those movements/operations. Little or no attention is on the start number. The start number is irrelevant to the challenge. The fact that attention is away from the start number and is placed elsewhere allows the opportunity to introduce a letter whilst attention is elsewhere. Once learners are used to the re-creating activity, it can be played again, only this time the start is not a number but a letter. Initially learners might react to a letter being placed in the grid rather than a number. However, if the familiar challenge of re-creating an expression is offered quickly, they recognise the activity is essentially the same activity they have been doing previously. The appearance of a letter has no bearing on their ability to complete the task and they soon accept the appearance of a letter (Hewitt 2014b). Grid Algebra has many computer generated activities and one of which is being given an expression and having to re-create that expression on the grid within a given time period. An example of this is shown in Fig. 7 where the task is to re-create the expression \( 2 \left( \frac{a + 9}{3} + 1 \right) - 6 \) within 30 seconds. I have found learners very engaged in such activities and the time restraint shifts attention even more onto what is relevant for
the task. This means attention is on the operations and the appearance of a letter becomes something which is quickly accepted.

Fig 7 A computer generated task where the expression has to be re-created within a time period

I will not go into detail here but there are other activities where the meaning given for letters can be developed. On an otherwise empty grid, the placement of a letter onto the grid still does not define the grid and the letter can take one of many values (according to which row it has been placed in). This gives a sense of letter as variable. A number can then be dragged from the Number Box and placed into the same cell as the letter, which will then define the value of that letter. Alternatively if a number is placed in the same cell as an expression generated from that letter then an equation such as \(3(k + 4) - 6 = 42\) is formed. This makes \(k\) an unknown and new challenges of trying to find the value of \(k\) can begin (more is said about the equals sign and solving equations later).

Parsing expressions

Standard notation of the four basic arithmetic operations can appear relatively complex for learners. It involves the use of division lines rather than a division sign, use of brackets, the absence of a multiplication sign, the particular positioning of numbers and operation signs’ and rules as to which operations have precedence over other operations. Standard notation has a vertical dimension to it as well as horizontal. Consequently the order in which it meant to be read can be a long way from the usual left-to-right used with reading an English text. The visual journey to consider the order of operations within the expression \(\frac{4\left(\frac{15}{3} + 6\right)}{2} - 8 + 4\) can be seen in Fig. 8. This is far from the usual left-to-right order and teachers as well as learners can have difficulty with correctly reading the order of operations (Glidden 2008). There are acronyms which are used by teachers to help learners come to know the order of operations. In the UK BIDMAS (Brackets Indices Division Multiplication Addition...
Subtraction) is commonly used and in the USA PEMDAS (sometimes said as Please Excuse My Dear Aunt Sally) is used to indicate Parentheses Exponents Multiplication Division Addition Subtraction. However, these are also fraught with difficulties as sometimes the order of M followed by D can lead to learners incorrectly thinking that multiplication always has to happen before division and not left-to-right as in the example of $24 \div 2 \times 3$. Or that addition must always take place before subtraction in an expression such as $10 - 2 + 3$.

Fig 8  An example of the visual journey required in order to read formal arithmetic notation

Booth (1984) carried out seminal research which identified many difficulties leaners had with notation. With regard to brackets, for example, she found that learners often ignored them when writing expressions themselves, sometimes due to feeling that the context of the problem they were working on would determine in their own mind the order of operations. Many learners also felt that the same value would be obtained anyway whatever the order of the calculation. Gunnatsson et al. (2012) explored using ‘useless’ brackets, which were additional brackets included in expressions beyond those which would be required under normal conventions, such as with $2 + (4 \times 5)$. These were used as a teaching tool to help learners carry out the correct order of operations. However, they found that in tests carried out with conventionally written expressions, this use of brackets was a hindrance to learners transferring from incorrect straight left-to-right reading of certain expressions to the correct conventional order. They felt that this could be due to the fact that learners were unclear when brackets were an essential part of the convention of reading expressions and when they were being used as an educational device.

I argue that a contributing factor to the difficulty of seeing order of operations within an expression is that the expression exists as a totality within a page. This contrasts with speech which necessarily happens in time; one word is said before another. There can be difficulties in describing order of operations within speech, but the awareness that time brings with it a natural order is utilised visually within Grid Algebra. The expression $\left(36 + \frac{12}{2}\right) - 4 + 18 - 2$ in Fig. 6 did not just appear all at once. Instead it was built up one operation at a time, with one movement necessarily happening before the next movement. Furthermore, the first movement is the first operation; the second movement, the second operation; etc. Consequently the expression is gradually seen to build in the order of the operations, so that the final expression has been created gradually over time and in the correct order of operations. Usually, when such an expression is written down, even if a learner watches a teacher or fellow learner write the expression, it is usually written left-to-right as much as possible, with the first number written being ‘3’. However, with Grid Algebra the operation – 4, for example, will appear before the 3 appears, since that subtraction takes place before the multiplication. This helps learners see the order of
operations within the final expression as it has been gradually built up one operation at a time in the order the operations are to be carried out.

The re-creating an expression activity is key in helping learners parse expressions. A learner has to look at the structure of the given expression, such as the one above, in order to decide how to move the start number of 36. Whatever movement they carry out results in the software feeding back the notational consequence of that movement. So, for example, if they felt that they should divide by two first and then add 12, then making the associated moves would result in the expression \( \frac{36}{2} + 12 \). This neutral feedback is informative for a learner to see whether they have the order correct since the visual appearance of \( \frac{36}{2} + 12 \) looks different to the equivalent part of the goal expression. This purely visual feedback is enough for them to judge whether they are interpreting the order of operations correctly (Hewitt 2014b).

Numbers can be entered into the grid by dragging them from the Number Box or entering them through an Expression Calculator. The latter allows an expression to be entered through pressing buttons similar to those found on a calculator. Fig. 9a shows a route drawn on the grid. The letter \( d \) has already been dragged from a Letter Box into the cell at the start of the route (it cannot been seen at present due to the route marker but if that cell is clicked on, the letter \( d \) will appear and that part of the route will temporarily disappear). The Expression Calculator has been put into the final cell of the route and the task is to enter into the calculator the final expression if \( d \) was to be taken on that journey.
Fig 9a and 9b (a) The grid where the letter d has already been placed in the start cell of this journey and the task is to enter the final expression; (b) How the notation appears on the screen after each button press.

Fig. 9b shows what appears inside the Expression Calculator as each button is pressed to create the expression $5\left(\frac{d}{3} - 2\right) + 15$. There are significant design features within this sequence. Firstly, the expression must be entered in the order of operations and not left-to-right. If someone attempts to start with the number 5 and then presses the $\times$ button, then the Expression Calculator is designed so that the letter $d$ is no longer available on the calculator. Thus they cannot continue. Pressing the Clear button suddenly brings back the letter $d$ as an optional button to press. So a learner is forced into starting with the letter $d$ since to start with anything else would make the letter $d$ disappear. This is a design choice as the route starts with the letter $d$ and now the learner is more likely to attend to the physical journey indicated by the marked route which helps them stay in touch with the order of operations.

After the $d$ button is pressed, the next button pressed is $\div$. However this symbol does not appear on the Expression Calculator screen. Instead, a line appears below the letter $d$ (see Fig. 9b for this and what follows regarding the entering of this whole expression). The fact that one symbol is pressed whilst a different notational symbol appears on the screen helps to associate one with the other. Pressing the 3 button next shows the 3 appearing below the division line. So, the Expression Calculator will show the conventional notation step by step as the operations are entered via the buttons.
The next button pressed is the subtraction button. The significance here is the position of the subtraction sign in relation to the current expression on the screen. The subtraction appears at the same vertical level as the division line and not, as is commonly done by many a learner, at the height of the expression on the top.

After the number 2 is entered, the multiplication sign is pressed. As with division, one symbol is pressed but a different symbolism appears on the screen. This time brackets appear. This sets up an association with the act of putting brackets round anything which is not a single symbol when that expression is to be multiplied. In this case it is going to be multiplied by 5 and on pressing 5 the number appears to the left of the bracket rather than to the right. So the Expression Calculator will show each symbolic part of the final expression at the same time as each button is pressed; associating mathematical operations with symbolic features and assisting seeing the final expression in the order of operations rather than left-to-right. I will mention other factors which assist the learning of order of operations later on.

**Equals sign**

The final topic I will discuss regarding difficulties learners experience concerns the equals sign. This sign is often viewed by learners as a signal to carry out a calculation and place a single symbol afterwards (Behr et al. 1980; Kieran 1981; Linsell and Allan 2010). This is exemplified with learners entering the number 15 into the box in the statement $7 + 8 = \square + 9$ (Warren 2003). The use of some calculators can assist this view as pressing a button with the equals sign on it produces the result of a calculation. This operational view of the equals sign does not tend to change between the ages of 11 - 14 and can also affect later performance in solving equation tasks (Knuth et al. 2006). As a consequence, learners sometimes have difficulty when presented with a statement where there is an expression on the right-hand side of the equals sign as well as the left-hand side, such as $7 + 8 = \square + 9$ mentioned above. Carpenter et al. (2003) suggested that there are three types of misconceptions which learners can have when trying to calculate the number which should go into the box: either *answer comes next* ($7 + 8 = 15$); *use all numbers* ($7 + 8 + 9 = 24$); or *extend the problem* ($7 + 8 + 15 + 9 = 24$). A more relational view of the equals sign would see learners view the equals sign indicating that both expressions either side of the sign are numerically equal. An example of this is the equation $28 ÷ 7 + 20 = 60 - 36$. However Cooper et al. (1997) found that about half of a group of 51 Grade 7 learners could not explain adequately what the equals sign meant in this context. The fact that so many learners have difficulty interpreting the equals sign with a relational meaning may be a consequence of the nature of the way learners are taught. In a comparative study, Li et al. (2008) found that there was a significant difference between Chinese and US Grade 6 students where a large percentage of US learners appeared to view the equals sign operationally whilst almost none of the Chinese learners did so. When comparing text books they found that the US text books rarely portrayed the equals sign relationally whereas this was very common with the Chinese text books. So, the nature of the experiences learners are offered may be significant in the way in which they view the equals sign.

So far in this paper I have not mentioned the equals sign in relation to Grid Algebra activities. The Expression Calculator does not have an equals button. It is used only as a means of entering an expression and does not carry out any calculations. In fact no tools carry out any calculations within Grid Algebra. More will be said later as to why this is the case. A sense of two expressions being the same comes from the fact that two expressions are in the same cell. For example, in Fig. 4 I talked about making a movement on the grid for the first time. The horizontal movement created the expression $1 + 1$ and this appears in a cell which previously had the number 2 in it. Looking again at Fig. 4 you can notice that in the bottom right-hand corner of that cell there is a ‘peeled-back corner’. This icon indicates that there is more than one expression in that cell and clicking on that icon will change having $1 + 1$ in view in that cell to seeing 2 instead. Another click on the icon will return it to $1 + 1$. In fact, if there are several different expressions within the same cell, each click on the peeled-back corner will cycle round all the expressions one at a time. This gives a sense that the ‘label’ for that cell could be any one of those expressions and hence links the expressions together as having something the same about them. The structure of the grid and the movements which can be made ensure that all expressions in the same cell are numerically or algebraically equivalent.
Fig 10  Different routes taken from the n cell to another cell with expressions linked by the equals signs

All expressions within a cell can be seen at the same time through the use of a Magnifier tool. When this tool is selected from the toolbox and dropped into a cell, a new small window appears showing all the expressions within that cell connected together through the appearance of equals signs (see Fig. 10). Here there have been four different routes taken from the n cell to the final cell, resulting in a set of four equivalent expressions. Whichever expression is currently on view within the cell is the expression on the left-hand side (LHS) of the top equals sign. All other expressions which appear in the cell are listed on the right-hand side (RHS) of equals signs in the order of most recently entered into the cell down to the expression which first appeared in the cell. However, the order can be changed easily by simply clicking on an expression within the Magnifier window; that expression will then become the expression on the LHS of the top equals sign (and be on view in the cell) with the others appearing on the RHS. By this means any particular stressing or ordering of expressions can be achieved. This ability to swap expressions from the RHS to the LHS and visa-versa helps to develop a relational view of the equals sign rather than an operational view. Also, within activities such as the one I describe below, the emphasis is on expressions rather than making a single number the focus of attention, this reducing the desire to carry out operations.
The mathematical notion of equivalence has an associated notion of making a journey from one particular cell to another particular cell. Some journeys are longer than others and consequently those resulting expressions involve more operations than others. Careful choices of journeys can assist learners in gained awareness of manipulation of expressions. Fig. 11 is an example of an activity I call *I go this way, you go that way* (I go horizontally, then vertically, whilst you go vertically, then horizontally). The resulting equation of \(2r + 6 = 2(r + 3)\) or \(2(r + 3) = 2r + 6\) (since they can easily be swapped round) creates an opportunity to work on the distributive law and factorising. The question: *why did it say plus three when I went my way but it said plus six when you went your way?* can be engaged with in two ways; either with reference to the algebraic notation or with reference to the physical journey. The latter allows learners to say that their journey had gone into the two times table first and so moving the three cells to the right meant doing three lots of two (as you add two each time in the two times table). The physical image of moving on the grid supports the mathematical awareness of what happens with the notation, the ‘3’ being doubled. The supportive image of the grid is important when someone is beginning to learn but eventually it is desirable for a learner to be able to multiply out brackets without the support of the grid. Consequently the design has a built-in facility to shift focus onto the symbols on their own. For example, one of the built-in computer generated tasks is based upon this activity of *I go this way, you go that way*. Learners have to enter in via the *Expression Calculator*, the expression for the alternative route. It gradually withdraws the support of the grid, initially withdrawing the grid whilst keeping the picture of the arrows for the two routes, and then withdrawing even that so that there is only the expression left (Fig. 12).
Fig 12 Expanding brackets task with support of the grid gradually withdrawn so that learners’ attention shifts from the physical journeys on the grid to the mathematical operations in the expression

The final design issue in this section relates to how different types of expressions can be brought together in the same cell. So far this has created identities such as $2r + 6 = 2(r + 3)$. However, an expression involving a letter can be created in a cell and then a number can also be entered into that cell, creating an equation with an unknown, such as $2r + 6 = 32$. This brings the possibility of working on solving equations, which is addressed later. The Letter Box allows different letters to be dragged onto the grid and taken on journeys. If different letters are brought together then equations such as $y = 3(x + 2)$ can be created. If there are no numbers on the grid then these letters remain variables. This is because, for example, if the letter $y$ is placed into a cell in the three times table when the rest of the grid is blank, then it can be any multiple of three. If a number is then put into a cell, then the whole grid is defined and each cell must be a particular number. This defining of the grid can also happen when no numbers appear at all. The design allows the possibility of more than one copy of a particular letter to appear in the cells. Fig. 13 shows a series of three screens. The first shows the letter $a$ placed in a cell and at this stage the letter is a variable (note the grid icon at the top-left is empty). The next screen in Fig. 13 has a second copy of the letter $a$ placed in a different cell. Using the convention that copies of a letter stand for the same number, there is just one possibility for the value of $a$ given that it appears in those two particular cells (note that now the grid icon is full of numbers, indicating the grid is defined and particular numbers must go into each cell). The final screen shows an example of two ways in which these letters can be brought together to create the ‘clues’ for working out the value of $a$: $2(a + 1) = a - 4$ and $a = 2(a + 3)$. Now only the number $-6$ would be accepted into either of the cells containing the letter $a$. The design is such that the first appearance of a particular letter can be placed in any cell, the second will only be accepted into cells where the forced value of that letter would be an integer, and any additional copies of that letter would then have to be placed in cells which would have that same numerical value.
Fig 13 Three screen shots where the second appearance of the same letter defines the grid (as indicated by the grid icon being full of numbers) and the letter shifts from variable to unknown. In the third screen shot the letters are brought together in two different ways, each of which acts as a clue to find the value of the letter

Theoretical perspectives

I will now discuss three frameworks which are central to the design of the software. These have been left to this part of the paper deliberately so that their influence on the whole design can be seen in the light of knowing many of the key functionings of the software. Two of these frameworks I developed some time ago and have been central to the design of Grid Algebra. These are *Arbitrary and Necessary* and *Subordination*. I will address the significance of these in turn below followed by a discussion of how the third framework, *Embodied Cognition*, has also influenced the design of the software.

**Arbitrary and necessary**

Arbitrary and Necessary (Hewitt 1999, 2001a, 2001b) divides the mathematics curriculum into those things which are social labels and conventions (Arbitrary) and those things which concern properties and relationships (Necessary). The first group of socially agreed labels and conventions include names such as *square*, *five* and *horizontal*. These words have become established over a long period of time and although there may have been good reasons at the time why each word was adopted, there is nothing necessary about the choice of each particular word. Indeed in another language different words are used. So there is nothing about a square which means it *has* to be called *square*. Likewise with conventions that the first number of (2,3) is the *x* co-ordinate and not the *y* co-ordinate, or that a whole turn is divided up into 360 when deciding a way to measure turn.

There are historical reasons why 360 was chosen that relate to the culture at the time (the Babylonians used a
Those things which are necessary have to be how they are, given certain arbitrary decisions. There is no choice. So, if we are working with the arbitrary choice of 360 degrees in a whole turn, then it is necessary that half a turn will have 180 degrees and reasons can be given for why this must be so. The mathematics lies within the relationship between a whole turn and half a turn rather than the particular numbers involved. Indeed a similar relationship will be argued when talking about radians as much as degrees. The necessary concerns properties and relationships and this is where the mathematics lies. The equation $2(r + 3) = 2r + 6$ involves many arbitrary labels and conventions in terms of the symbolism and notation used. However, the relationship is about operations: that if I add three and then double, it gives me the same numerical result as doubling and then adding six. What number I started with is irreverent. This remains true no matter how I might express it notationally. So, suppose a learner were to work on a problem where the task is to find a general rule, such as $2(r + 3)$. However, instead of writing this, they write $r + 3 \times 2$. What they have written is ‘wrong’ but what is of importance is to find out what they are wrong about. If they think that the rule is that you take $r$ and add to it $3 \times 2$, then they are wrong mathematically but right notationally. If they think the rule is $r$ add three and then multiply by two, then they are right mathematically but wrong notationally. The difference is significant for a teacher. In the first case the learner’s awareness of the mathematical situation needs to be addressed; whereas in the second case it is only a matter of helping them adopt a convention about how the rule is written down. The first concerns the necessary and the second, the arbitrary.

Due to its nature, learners cannot know the arbitrary without being informed (Hewitt 1999). So at some point a learner needs to be informed of the names and conventions which are used within the mathematics community. So the role of a teacher is to inform and assist learners in adopting and using these names and conventions. A teacher’s role is quite different for what is necessary. Since the necessary is about properties and relationships, those things for which there are reasons why they have to be how they are, learners can work out these things for themselves. As such a teacher’s role is more concerned with offering appropriate tasks and challenges and using questioning skilfully to assist students in educating their awareness. So a principle I have developed is one of providing the arbitrary whilst challenging learners to become aware of what is necessary.

Grid Algebra is designed with this principle in mind. The arbitrary is the notation; how expressions are written. Grid Algebra provides the standard notation and a learner does not have to know the conventions of arithmetic/algebraic notation in advance of using the software. The notation appears as a consequence of a movement on the grid; or the notation appears as the operations are entered into the Expression Calculator. At the same time no calculation is performed by the software, nor is any solution to an equation or re-arrangement of an expression carried out by the software. Calculations, solving equations and re-arranging expressions concern the necessary and as such the software offers challenges related to these within the built-in computer generated tasks. In addition, the ‘Interactive Grid’ (which allows teachers and learners to work with the grid in whatever way they wish) offers opportunities for a teacher to set up challenges and use questioning to challenge learners in a way where they are likely to gain new awarenesses about number, operations, expressions and equations (a few of which have been mentioned above). Learning about inverse, order of operations, equivalence of expressions, distributive law, solving equations, becoming aware of when a letter is an unknown and when a variable, etc., can all be achieved through questioning and challenging learners with appropriate Grid Algebra tasks. Whilst learners do this, the software provides what needs to be provided: the notation.

There is a significant advantage to a computer providing the standard notation rather than a teacher. Learners are used to adapting themselves to a technological environment where they have to get used to working with arbitrary things, such as icons, button pressing at particular points and combinations of buttons required. I have found that learners may question initially why something is written how it is, such as why division is written as
a fraction rather than with the ÷ sign or why a multiplication sign is not used. However, there seems to be a quick acceptance that this is how the software writes things even though it is quite different to what they are used to seeing in their previous years of mathematics lessons (Hewitt 2012). I have sat in many mathematics lessons where teachers attempt to justify why they are not using the multiplication sign and rarely do I feel that the learners are convinced. The question why? is very appropriate for the necessary but not for the arbitrary. It can lead to a distraction with learners feeling unclear about something before any mathematics is addressed. Learners are often aware of the fact that a teacher has choice and so wonder why the teacher chose to write it in a different way to what which they are familiar. With technology, there is a sense that getting to know the technology will always involve a number of arbitrary things. So learners are more likely to ‘accept and adopt’ with technology.

Subordination

When we learn something well we are often not consciously aware of using that learning when we meet a challenge that requires its use. When I hear the door bell, I go to the door and see who is there. In doing so, I do not think consciously about what is required from my muscles in order to walk; I do not think about the fact that some dexterity is required to open the door (my door involves pressing a button with my thumb whilst rotating a small lever at the same time). What we use in order to achieve tasks are the things which are being practised. So my typing becomes increasingly fluent because I practise typing not because I want to practise per se, but because there are things I want to write. Dewey (1933, p.15, his emphasis) commented that “the nature of the problem fixes the end of thought, and the end controls the process of thinking”. As such it is the task which I want to achieve which can determine what it is that I will end up practising.

One feature of these examples is that attention is placed upon achieving the end task and what is practised is something for which I might already have significant fluency. This is sometimes a feature of when fluency has already been achieved; I do not need to pay attention to it. It is often different when someone is learning something; then attention is often focused on that which is to be learned. However, I argue that the notion of staying focused upon the end task whilst learning what is required to achieve that task, can be a powerful way in which someone can become fluent rather than just practising existing fluency. Fig. 14 gives a diagrammatic image for subordination (Hewitt 1996) and I use this term when:

(a) a skill or knowing, A, is required to achieve a task, B;
(b) the task, B, does not require A in order for it to be understood;
(c) a learner can observe and judge the relative success of their attempts at achieving task B;
(d) a learner can adapt and modify their use of A whilst trying to achieve B.

The first of the above criteria is that A is required in order to achieve B. This means that there is not an alternative means of achieving B. So, in a teaching context, that might involve the creation of carefully designed rules of what is and is not allowed within an activity. Or with technology, the software is designed so that a learner only has A available to use.

The second and third aspect means that even if a learner knows little about A, they can still fully understand what the task is and would be able to judge the extent to which they have been successful at this task.

The final aspect means that they can learn from the relative success they are currently having on task B and adapt and modify the way they are using A in order to see whether they can gain better success. Thus, feedback is crucial here. It is seeing the consequences of your actions which inform how your actions are modified or adapted in the future. It is this that drives the learning of A. Through attending to the consequences of using A, rather than A itself, a learner becomes more skilled with A. For example, I may use a hose to wash my car and when I turn on the water supply I can see whether the water is indeed going over my car. If it is not then I learn about how to use the hose not by looking at the hose but by looking at the car and where the water is going in relation to the car. So attention is on the relative success with the task. This is a relatively unusual feature inside a mathematics classroom as usually attention is on whatever it is that is being learned. With the use of subordination, attention is on the consequences of the current use of that desired learning upon a specially created task.

A relatively obvious, but significant, aspect of the design of Grid Algebra is that the notation is written in a standard form and not in a single line form such as \( \frac{36}{2} + 12 \). There are two significant aspects to this. Firstly, learners meet and have to work with standard notation and more about this is said below. Secondly there is an element of what diSessa (1985) described as naive realism, which is when a learner feels that objects are the visual representations they see on the screen. When a learner moves from one cell to another they can feel as if they are working directly with the mathematical expression since the expression is being moved as part of the cell. So a learner can feel as if they are directly changing the mathematical expression, such as writing “+ 2” onto an expression, when in fact they are just moving a cell to the right. This sense of working directly with the mathematical notation is important not only because it is what learners will be doing later when working in more traditional pen and paper contexts, but also because when making movements on the grid they can have intentions of changing the notation in particular ways whilst having fed back from the software the actual effect on that notation through the actions they took. This relates directly to the notion of subordination where attention is on the consequences of intended actions. Consider one activity which involves an expression, such as \( \frac{2(y-9)+12}{3} + 4 \), in a cell in an otherwise empty grid. The task is to find out where the letter \( y \) must have been placed in order to have created this expression through movements on the grid. This will require them to find the inverse journey. The challenge of finding where \( y \) must have been placed can be understood by learners even if they do not know initially how to go about solving this. They know the end game even if, at present, they do not know the means to achieve that end. Thus they are forced to pay attention to the symbolic notation and consider inverse in relation to that notation, as this is all they have to work with. This meets the first two conditions for subordination.

The third condition is that a learner can judge the relative success of, in this case, making an inverse journey to get back to where \( y \) must have been placed. The design of the software at present would mean that any further movement of that expression would add on an additional operation to the expression. For example, if a learner wanted to do the inverse of divide by two then they would drag it down in order to multiply by two. However, this would result in a new multiplication occurring in the new expression: \( 2 \left( \frac{3}{3} \right) \frac{3}{2} + 4 \). This would not be particularly clear feedback for the learner that they had ‘undone’ the division by two. So an Inverse button is available in the toolbox. When this is pressed, any movements of an expression will result in the usual extra
operation included within the resultant expression, except when the inverse of the last operation in the expression is carried out. In that case the last operation is cancelled out and will not appear in the resultant expression. This means that the movement to multiply by two will not result in the expression above but will result in the expression \( \frac{2(y-9)+12}{3} + 4 \). So a learner can judge their relative success through seeing whether their movement does cancel out an operation and simplifies the expression. So a sense of getting back to the letter \( y \) has a meaning both in terms of taking physical steps on the inverse journey back to where \( y \) was, and also in terms and taking mathematical steps to reduce the expression back so that only the letter \( y \) appears. The nature of the feedback is crucial. For example, a learner may originally think they want to undo the \(+4\) within the expression \( \frac{2(y-9)+12}{3} + 4 \). If they do this by dragging the expression four cells to the left they will see the resultant expression \( \frac{2(y-9)+12}{3} + 4 \). This will not give them a sense of having simplified the situation and they can also note that they do not succeed in cancelling out the \(+4\). So feedback is offered which does not explicitly say to them no this is wrong but does give them the consequences of their actions (Hewitt 1994) from which they can judge whether they are successfully working towards the end goal. Typically, this will result in them realising that they need to do something else instead and explore multiplying by two first. This will result in an expression which will confirm that they are getting closer to \( y \) (both in terms of the visual expression and also in terms of the physical journey). This design feature relates to what Edwards (1998) describes as interpretable feedback where expectations are compared with actual outcomes and means that both the third and fourth conditions for subordination are met. So through such tasks, a learner can start off not knowing very much about inverse operations and their order, but can end up learning and becoming quite proficient with finding the inverse operations through having to subordinate this continually to the tasks at hand. They come to learn without the need for explicit instruction but through continual practice combined with neutral yet highly informative feedback.

The role of subordination is present within much earlier work with Grid Algebra, including the task mentioned above of trying to re-produce an expression. It is through subordination that learners become familiar and comfortable with formal expressions and learn the order of operations within such expressions, and this is one of the main aims of the software. So often learners are able to find rules in mathematical problems but have difficulty expressing those rules in standard notation (Swafford and Langrall 2000). Grid Algebra offers a way of starting with standard notation and learning to interpret it correctly through attending to the consequences of the reading of that notation rather than through explicit instruction. This mirrors how that notation will be used once such learning has become automatised with the focus of the learning being on the completion of given challenges which require the use of that learning rather than on the learning of order of operations as an isolated piece of the curriculum. Thus learning through subordination reverses Rousseau’s (1986, p. 90) edict of “before you can practise an art you must first get your tools”; with subordination, learning the tools happens as a consequence of wanting to practise your art.

**Embodied cognition**

Varela et al. (1993, p. 27) challenged the idea of considering mind and body separately and used the term embodiment to mean “reflection in which body and mind have been brought together”. As such cognition is not seen as something which is separate to “living bodily experience” (Núñez et al. 1999, p. 49). Thinking involves far more than solely a mental activity but involves verbal articulations and physical movements, often in the form of gestures (Radford 2011). There has been much interest recently in the role that gestures play in learning mathematics (Arzarello et al. 2009; Cook et al. 2008; Edwards 2009). However, the notion of embodiment goes beyond the idea of a learner using their body whilst learning. It is a statement about the fact that all learning, whether cognitive or physical, is rooted within deep experiences which are embedded within us in such a way that it makes no sense to separate out the mind and the body. Johnson (1987) offers the example of feeling physically balanced, which is a fundamental bodily experience which we all have experienced in one way or
another. This is a deeply important sense which relates to other experiences of whether things ‘feel right’ or whether something is too hot, fast, sharp, noisy, etc. This sense of feeling balanced is also prevalent within mathematical notions such as the use of the equality sign, approximation, equivalence and symmetry. It is fundamentally present within us all as we engage in all sorts of activities, no matter the nature of those activities. I offer here the example of inverse which is another deeply rooted notion embedded through many experiences throughout our lives. If a child moves something and is told to put it back, the movement the child makes is the same distance but in the opposite direction. We have all done this with our arm when pointing and returning to our previous position, we do it when we are walking from one room to another and then returning to the original room. The embodied sense of inverse is available to us all when encountering mathematical tasks as much as with any of the physical tasks mentioned above. Within Grid Algebra the number 4, for example, can be moved three cells to the right and then moved back to its original position. This physical act of returning back to where we started is one which is known at a deep level to us all. That does not mean that this is something which a learner might consciously be aware of when engaged in such activity. The fact that it is so deep rooted may, in fact, reduce the likelihood that we pay conscious attention to it. However, we cannot help but bring with us these perceptual-conceptual primitives, or image schemas (Johnson 1987; Núñez et al. 1999) to whatever tasks with which we engage. So a learner will bring with them their sense of inverse and making such a journey will fit in with the idea that the second movement was the inverse of the first and that they have returned to the same place. This is supported symbolically with the equation $4 + 3 - 3 = 4$ reflecting both the inverse nature of the operations and also the fact that they are back at the number 4, where they started. So $+3$ and $-3$ can be felt in a bodily sense as inverse and seen as inverse operations within the resulting expression. The perceptual-conceptual primitive of inverse supports the engagement in what is happening regarding the mathematical operations and symbolic notation.

Another design feature involves the physical movements themselves. Sometimes a learner wants to move diagonally from one cell to another. However, only horizontal or vertical movements are possible. This is so that there is a direct relationship between one physical movement and one mathematical operation. Even if someone wants to carry out two consecutive additions, for example to move the letter $x$ in row 1 to produce $x + 3 + 1$. They have to click on the mouse and drag the letter $x$ along three cells, then they must release the mouse button and click again to pick up the new $x + 3$ to drag it one more cell to the right. Thus to create two operations involves making two distinct physical movements, so that someone feels two movements when two operations are carried out. This helps establish a close link between physical acts and the resultant mathematical symbolism.
A journey from \( x \) to create an equation and an inverse journey along the same route which solves that equation

Many journeys, such as walking from home to a local shop, involve multiple changes in direction and we are all used to the fact that in doing the inverse journey along the same route, involves a reverse order of those directions. So the notion and significance of order is also wrapped up in the perceptual-conceptual primitive of inverse. Fig. 15 shows the letter \( x \) being taken on a journey involving multiple directions to create an expression.

The number 56 is then put into the same cell as the expression to create the equation \( 4\left(\frac{2x-12}{6}-1\right) = 56 \). Then the embodied sense of inverse takes the number 56 and moves it back along the same route but in the opposite direction to arrive back at the letter \( x \). The design of the software is such that the intuitive sense of carrying out the inverse journey physically along the same route is reflected symbolically by carrying out the inverse operations in order to solve the equation, resulting in \( x = \frac{6\left(\frac{56}{4}+1\right)+12}{2} \).

diSessa (1985, p. 4) spoke of a spatial metaphor, wanting “to tap the well-developed pool of knowledge about space that humans already possess” within the design of educational software and Grid Algebra does this in terms of our experiences of making journeys. The design of the software allows a teacher to “teach from embodied understandings” (Davis 1996, p. 239) where learners can intuitively get a sense of how to solve an equation as it runs alongside their intuitive notions of order and inverse. The solving of an equation feels like a natural process rather than a set of rules they have been told to do and try to remember. The software allows gradual fading of support, as in Fig. 12 so that learners end up solving equations with only the symbolic equations to work with, mirroring what they are required to do with pencil and paper.
There are other ways, which I will not elaborate here, where the design of Grid Algebra utilises other primitives to support mathematical activity. Equivalence, for example, builds upon the primitive of knowing that there is more than one way to go from one place to another. The equivalence is in the fact that alternative routes all start and finish in the same place. The resultant expressions created in Grid Algebra of such routes will result in equivalent expressions. The notion of ‘simplifying’ connects the idea of choosing a simpler route from all the possible routes from one cell to another. Such a ‘simpler’ route results in a ‘simpler’ expression.

Final remarks

The design of Grid Algebra has been based upon the frameworks of arbitrary and necessary, where only the arbitrary notation is produced by the software whilst the necessary mathematical properties and relationships come from engagement with carefully designed activities. The design stresses attention on operations rather than numbers through the dynamic movement which is possible on the grid. The embodied nature of carrying out physical movements is associated with carrying out mathematical operations through the consequences of movements being fed back in terms of mathematical operations written in standard notation. The fact that standard notation is the only thing which is created on the grid, means that this is subordinated to the challenges offered such as re-creating an expression or trying to solve an equation. The notion of subordination means that learners become confident and fluent with standard notation relatively quickly. Indeed, learners relish making quite complex expressions, involving 10 or more operations, whereas teachers are traditionally more inclined to use much simpler expressions. Learners also show confidence in dealing with these complex expressions. The bodily experience all learners have had with making journeys, and then making return journeys, means that these life experiences support the new awareness of identifying which mathematical operations are involved in solving equations, and in which order. As such, much of this early algebra learning can become intuitive. Some of the well-established difficulties that learners often experience with algebra have been considered within the design: expressions are treated as much as objects as they are as processes; the equals sign is not treated as an indicator to provide a single numerical answer; the parsing of expressions is quickly learnt through the subordination of those expressions to carefully designed tasks; and due to attention being on operations, a letter can be introduced and accepted with relatively little fuss.

All metaphors and analogies used as pedagogical tools will have their limitations as well as strengths. Wheeler (1996, p. 324) said that "a pedagogical device can be too noisy, too full of distractions, to achieve its objective." Indeed, with Grid Algebra there is ‘noise’ and the software does have limitations. I will list here some of these:

- Students need to remember with which direction each operation is associated (although I feel it takes little time for learners to become familiar with this);
- The number of rows dragged down in order to multiply an expression by two, for example, varies according to the row in which the start expression appears;
- There was a pedagogic choice of designing the grid so that only whole numbers are involved. Although the reasons for this are significant pedagogically, it means that students do not experience working with expressions which have fractional or decimal numbers;
- Not all linear expressions are possible within Grid Algebra. For example, $3x + 2 - x$ is not possible. I did consider ways in which such expressions could be created but I felt that this created far too much extra ‘noise’ to be worth the educational benefit it might bring;
- Expressions with powers are not possible;
- Although equations with $x$ on both sides are possible to create, Grid Algebra does not offer a way of operating directly upon either of those letters.

So Grid Algebra has many limitations. However, it also offers a highly intuitive way in which learners can become confident with many keys aspects of beginning work on algebra, including:

- becoming familiar and confident with formal notation;
- interpreting order of operations within relatively complex expressions;
- becoming used to the inclusion of letters in expressions and substituting values for those letters;
• multiplying out brackets and factorising expressions;
• generating equivalent expressions;
• working out inverse operations for a given expression;
• solving a sub-set of linear equations.

Many of the above come with a sense of generality, where learners’ developing awareness goes beyond the limitations of the precise expressions possible within Grid Algebra. For example, I have found that equivalent questions given within a pencil and paper environment were answered just as successfully as those completed within a Grid Algebra environment (Hewitt 2014a) and that learners were able to work successfully with linear expressions which were not possible within a Grid Algebra environment. The generality, for example, of carrying out inverse operations can be utilised with an $x$ in the equation $4x + 12 = 16 + x$ to produce $4x + 12 - x = 16$ just as it can with the 8 in $4x + 12 = 16 + 8$ to produce $4x + 12 - 8 = 16$. So, although the former manipulation cannot be done within the Grid Algebra environment, the generality learned within the environment carries over to situations outside the environment. It is also worth noting that any resource designed to be helpful for certain mathematical notions cannot be helpful for all of mathematics. As diSessa (1985, p. 3) suggested, there is a “possibility, even the necessity, of using different models for different purposes and of a gradual shift in the kind of model employed as a user becomes more experienced”. Grid Algebra provides certain key foundations for algebra in an intuitive way. Those can then be applied within new learning situations, possibly with the use of different models to develop further learning of algebra.

This paper has identified some of the key ideas upon which Grid Algebra has been designed. Whilst accepting that there are educational limitations contained within the functionality of the software, the design allows learners to engage with the beginning of algebra in a more intuitive way than is common within more traditional approaches to this ‘hard to teach’ topic in a way which also tries to redress some of the known difficulties many learners experience when learning in a more traditional way.

Several issues come to mind from the process of designing Grid Algebra, some specific to the software and others more general. I will finish with these questions:

Was the decision to restrict expressions to include only whole numbers a correct one? Rather than a move straight from the second to the third row not be allowed, the notation consequence could have been expressed as: $\frac{3}{2}(expression)$. There is a balance between opening up new possibilities for expressions and not wanting some notation to appear with which a learner might struggle to relate.

There could be ways in which a general, nth, row is shown and would allow expressions such as $2n + 3n$ by starring with 2 in row 1, dragging down to the nth row, and then moving three cells to the right. This would allow a letter to appear more than once within an expression but adds a whole new complexity in terms of the visual environment as well as rules as to what can or cannot be done between other rows and the nth row. This is an example of where a designer needs to consider whether something which is technically possible is educationally desirable. Metaphorical images and resources for mathematics can offer support for a learner but what equally is important is that the learner is learning mathematics and not just learning the resource. As such a learner needs to see the mathematics through the resource and not just see the resource. As such a designer needs to ensure the resource never gets so complex that a learner can no longer see through it to the mathematics.

How might a teacher work with resources which respects the tenet that learners are only informed of the arbitrary whilst they work on what is necessary?

What other educational resources can tap into the notion of embodiment so that learners engage with mathematics in an intuitive way?

What other technological environments can employ the notion of subordination in order to assist learners in gaining fluency with certain areas of mathematics?
References


Radford, L. (2002). The seen, the spoken and the written: a semiotic approach to the problem of objectification of mathematical knowledge. *For the learning of mathematics, 22*(2), 14-23.


