Case Studies in Integrating the Interactive Whiteboard into the Secondary School Mathematics Classroom

Roger Duke *Janelle Pollard*
rduke@itee.uq.edu.au janellep@itee.uq.edu.au
School of Information Technology and Electrical Engineering
University of Queensland, Australia

Abstract: Interactive whiteboards extend the general interactive nature of data projectors. With this merger has come several significant software design issues. Software can be used to demonstrate concepts, to enable complex calculations and even to teach techniques, but the technology with the right software can enable the teacher to do much more. Mathematics educational software can both guide secondary school students to a deeper understanding of mathematical structure, and enable students to move away from the 'tools and techniques' view of mathematics and start developing mathematical thinking. The computer becomes an experimental instrument whereby ideas can be explored and relationships discovered. For such goals to be fully realised, the technology needs to be successfully integrated into the mathematics classroom. A first step in achieving this is for the teacher to work closely with the software engineer when planning and creating classroom lessons. In this paper we illustrate through case studies the role software can play in personalising the interactive whiteboard for the individual teacher so that it becomes an integral part of the teacher's lesson plan. By developing generic software that can be readily modified to meet the needs of individual teachers it should be possible to greatly increase both the take-up and effectiveness of the technology.

1. Introduction

The use of an interactive whiteboard in the mathematics classroom has opened up the possibility of lessons involving teacher, students and computer in an active partnership. But this raises several closely related issues.

- What is the most appropriate and effective mathematics-oriented software to run on an interactive whiteboard?
- How can teachers be encouraged to make full use of this technology?
- How can the technology be fully integrated into the mathematics classroom?
- What is the nature of the teacher-students-computer partnership and how can this partnership be fostered and developed?

In this paper, two case studies will provide some insight into these issues and illustrate the potential of the interactive whiteboard to improve the teaching and learning of mathematics.

1.1 Background and Related Work

The British Ofsted report on the effect of the government's ICT initiatives in secondary mathematics (see [22]) does not paint a very positive picture:

> "The effect of government ICT initiatives on the quality of teaching and learning in mathematics varies considerably among schools. Overall good practice remains uncommon."

The report goes on to observe that "many mathematics teachers remain unaware of the potential of specific software and tools".

Research in Australia has reached similar conclusions (see [10] and [19]). The British Ofsted report cited above, as well as a recent report by the Queensland government (see [28]), clearly illustrates that a particular emphasis of those in government working to foster the integration of ICT into the mathematics classroom is to encourage the teacher to carefully study, perhaps with assistance from a teacher-support unit, the available mathematics software packages so as to be able to use these packages in the classroom. Although the currently available off-the-shelf software packages are excellent, this software is not necessarily ideal for interactive whiteboard use. Furthermore, these general packages are often large and require extensive and detailed study by the teacher and students before they can be used interactively in the classroom. One problem with a pre-prepared software package is that it is usually not possible, particularly for a non-IT expert such as the average mathematics teacher, to customise the software: you take the package as is or not at all.

Teachers who have successfully made the transition to the later phases of integrating ICT into their teaching (i.e. the 'adaptation' or 'appropriation' phases described by Dwyer et. al. in [7] and [8]; see also [18]) have the knowledge to adapt off-the-shelf packages to meet their needs (for case study examples see [7], [9] and [12) and tend to choose more flexible and generic software packages (see [15] and [7]). Our research experiments reported in [26] (and the work of others reported in [20], [29] and [21]) have shown, however, that teachers at the 'entry' and 'adoption' levels of integrating ICT into their teaching (see [7] and [8]) are unlikely to adopt current software packages unless the packages are close to their current teaching style.

Over the past 3 years we have been involved in a project to explicitly address this issue by working in close cooperation with teachers to create individualised software fitting the teacher's particular teaching plan and style (see [24], [25] and [26]). This work has revealed a need to make software content specific yet able to be tailored to the teacher. The overall aim of this project has been to create a guided process to enable software engineers and educators to combine the best of their resources to form innovative yet practical mathematics educational software. It is hoped the eventual outcome of this project will be a common language of communication for both software designers and classroom mathematics teachers, enabling the development of sound and practical platforms for the creation of educational software designed specifically to meet the needs of teachers.

In part, this paper is a continuation of this goal. Of particular interest is an exploration of how the interactive whiteboard, together with appropriate software, can help stimulate the development of the secondary school students' mental images and enable them to explore beyond the 'tools and techniques' aspects of mathematics with its emphasis on practical applications. Whilst we advocate a problem-based learning approach, our emphasis is on thinking mathematically and less on mathematics as a tool for solving practical problems.

The possible role of the interactive whiteboard in teaching mathematics has been investigated by a number of authors. Greiffenhagen in [13] and [14] explored their use in teaching mathematics to secondary school students, while Perks in [23] looked at the implications of their design and use. Observations on the use of interactive whiteboards in British schools have been recorded for the Teacher Training Agency by The Mathematical Association (see [31]), and an excellent resource when searching for existing software is Curriculum Online (see [5] which specifically separates out software for the interactive whiteboard).

The software engineer can play a central role in personalising the use of the interactive whiteboard for the individual teacher so that it can become an integral part of the teacher's lesson plan. The case study in Section 2 gives an example of such software together with a lesson plan aimed at exploring the concept of real functions and their graphical interpretation. One of our aims is to develop underlying generic software structures that can be readily extended to meet the software needs of individual teachers. It is our belief that this can greatly increase both the acceptance and effective use of the technology.

We believe that the interactive whiteboard can also help students discover the process of thinking mathematically. It is often the case that studying secondary-school mathematics only indirectly leads to the student experiencing and exploring the pleasures and rewards of thinking mathematically. The case study in Section 3 gives an example of how the interactive whiteboard can be applied to directly foster mathematical thinking.

2. Case Study: Exploring Real Functions

This case study arose from discussion with practicing UK secondary school mathematics teachers about software they would like for their interactive whiteboards. One particular request was for software that would empower the teacher to explore the notion of a real function, and investigate the way real functions can be represented. At first this request seemed suprising, as there are many excellent graph drawing packages (see for example [1], [2], [6], [11] and [17]) in addition to widely available graphical calculators (see for example [3], [4] and [30]). The software and lesson plan that resulted from further detailed discussion is explored below. The software was designed specifically for the interactive whiteboard and enables a high level of interaction between teacher, students and the whiteboard.

The lesson plan requires the teacher to select a function from among a collection of pre-programmed functions. This collection includes linear, quadratic and trigonometric functions, as well as functions like $y = |x|$ and discontinuous functions like $y = 5$ for $x \geq 0$ but $y = -5$ for $x < 0$ (a Java version of the software can be downloaded from www.itee.uq.edu.au/~rduke/fun.zip).

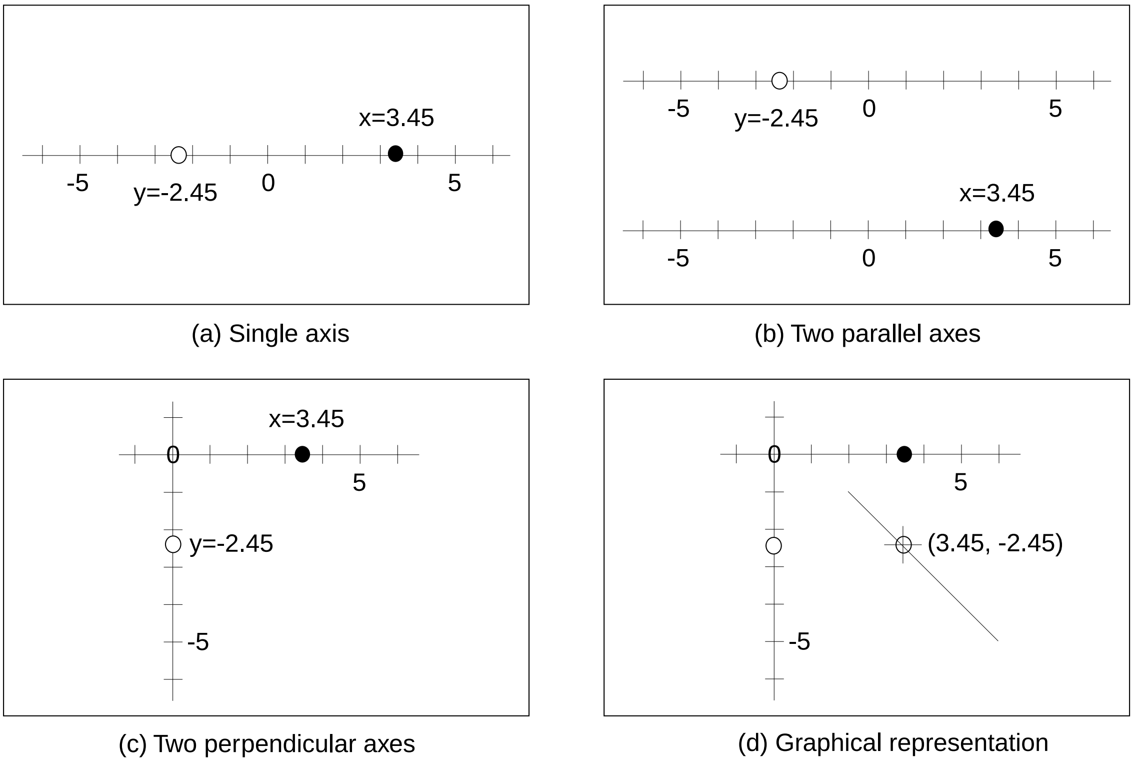

Figure 1 Function Representations

In the example illustrated in Figure 1, the linear function $y = 1 - x$ has been selected by the teacher. Although the teacher is aware of the function chosen, this information is not at this point available to the students. The screen initially displays a single axis, as shown in Figure 1(a). The point denoted by x can be dragged along the axis using the whiteboard pointer; as it moves, the displayed value of x changes as appropriate. At the same time, the point y also moves so as to maintain the functional relationship between x and y. Effectively, the software is representing the function as a set of ordered x, y pairs. In addition, when x is dragged along the axis the relationship between the motion of x and the resulting motion of y gives significant clues about the nature of the function.

The students are asked to use the software interface to investigate the function and to conjecture as to the functional relationship between x and y. In an ideal environment there would be additional computers in the classroom so the students can explore the function in small groups but then come to the whiteboard to discuss, explain and test their conjectures. The teacher may give clues, e.g. by telling the students that the function is linear.

Instead of the point x and the related point y being on the same axis, the teacher can select to display the function by placing x and y on different (parallel) axes (see Figure 1(b)). This is a perfectly valid alternative representation but contains no extra information.

The teacher can further select to display the two axes drawn perpendicular; again this representation is just as valid and contains no extra information (see Figure 1(c)). In each case, the x point can be dragged along its axis by the whiteboard pointer and the y point moves so as to maintain the functional relationship.

Given the perpendicular axes, the teacher can select the display to go one step further. As the x point is dragged, the ordered (x, y) pairs that result can be recorded as positions in the Cartesian plane (see Figure 1(d)). Effectively, as the x point is dragged, the graph of the function is drawn. This pictorial representation enables all the ordered pairs that define the function to be seen in one (graphical) diagram. The shape of this graph gives us important information about the function which is not easily deduced from looking only at the ordered pairs as in Figures 1(a) to 1(c), and clearly illustrates the importance of representing a function by its graph.

2.1 Discussion

The exploring-real-functions software discussed above raises several mathematical issues that can be investigated by the teacher and students. The first and most obvious is the idea of a function capturing a relationship between values: the value of x uniquely determines the value of y. However, representing a function just as a set of ordered pairs makes it difficult to fully comprehend the nature of the function or to compare the similarities and differences between functions. The important concept is that the values of x and y can be represented as positions on the real line, and this suggests several alternative but closely related ways of representing a function, leading inevitably to the notion of the function's graph.

In the process of discovering these representations, the students can be encouraged to formulate and test conjectures, and to develop strategies for determining the nature of a function (e.g. if it is linear, quadratic, trigonometric) given the various representations. In particular, the dynamic nature of the software enables the relationship between the movement of x and the resulting movement of y to be explored, and to ultimately relate this to the shape of the graph of the function. Specifically, the direction y moves in response to the movement of x indicates if the slope of the graph is positive or negative, and the speed of y relative to the speed of x indicates the magnitude of the slope. This level of analysis would be almost impossible without the animation that the software offers. Furthermore, the software is designed to be facilitated by a teacher; it is not intended for completely independent student use. The software is intimately tied in with the overall lesson plan and both the software and the lesson plan need to be developed together.

When demonstrated to other teachers, there were many requests for individual customisation. One teacher wanted the software to explore the basic function concept for complete beginners. Another wanted to use the software to explore the notion of function composition, with an extra point z having a functional dependence on the point y.

As functions and their graphical representations is a core concept in secondary school mathematics, it can be expected that the ideas and techniques developed in the software would be widely reused when customising software for individual teachers. If such customisation is to be achieved, it is important that the teacher and software engineer work closely together; we believe that being able to customise software is crucial if we are to encourage the teacher-uptake of the technology.

3. Facilitating Mathematical Thinking

The importance of developing mathematical thinking has long been recognised (see [27 and [16]). In this section we explore software that enables the teacher to facilitate mathematical thinking through the use of an interactive whiteboard. In order to focus on mathematical thinking, as distinct from practical mathematical applications and techniques, it was decided that the software should tackle a problem that requires minimal formal mathematical background. Ideally, the problem should be relatively straightforward to solve, but in the process suggest other mathematical ideas and concepts. In particular, it is important that solving the problem is not seen by the students as a competition aimed at testing and rewarding only the quickest and brightest.

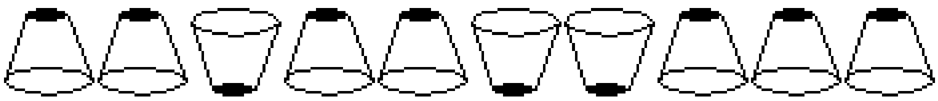

Figure 2 Flip the Glasses

Figure 2 illustrates the interface of the software for a simple puzzle. Glasses are arranged in a line; some are upright and some are upside-down. The aim of the puzzle is to end up with all the glasses upright, but by doing nothing more than repeatedly selecting two adjacent glasses and flipping (turning over) them both. At each stage the two adjacent glasses to be flipped are selected using the whiteboard pointer. The software interface is such that at any time either the glasses can be completely restored to their original configuration so that the same puzzle can be attempted again, or a new configuration can be created randomly. The number of glasses in the line can be set to be anything between 1 and 20.

It turns out that the puzzle can be solved if and only if the number of glasses upside down is even. If this condition holds, one strategy for solving the puzzle is to select for flipping those two adjacent glasses with the property that the glass on the left is the left-most upside-down glass in the line, and continue with this selection policy until all the glasses are upright. The lesson plan for the students is to discover the necessary and sufficient prior condition, to develop a strategy for solving the puzzle when this prior condition is satisfied, and to formulate a reasoned argument to justify why the puzzle can be solved if and only if the condition hold.

3.1 Discussion

As this puzzle is quite straightforward, a broader lesson plan is required if the teacher wishes to raise and investigate related underlying mathematical issues and to help students develop mathematical thinking skills. The computer, through interaction with the whiteboard, is simply acting as a resource and does little more than enable students to experiment and explore. Nevertheless, it is an important resource because it would be difficult to perform the investigation with actual glasses. If, as for the case study in Section 2, there are additional computers in the classroom, the students can explore the puzzle in small groups but come up to the whiteboard to discuss and demonstrate any conclusions. One desirable outcome of software such as this is that the students discover that mathematics is not a solitary experience but is enriched by sharing and communicating with others.

The first mathematical issue is that of specialisation: investigate the problem when there is just 1 glass, then 2 glasses, 3 glasses and so on. Our experience with trialing the software with year 9 students (13-14 years old) is that they become very enthusiastic and rush to solve the puzzle; it requires careful guidance from the teacher to get the students to take a systematic approach. The students need to be encouraged by the teacher to formulate and test conjectures until there is consensus that the parity of the number of upside-down glasses is the crucial factor that determines solvability.

A central issue is why a necessary and sufficient condition for the solvability of the puzzle is that the number of upside-down glasses be even. The lesson plan here is not for each student to write a formal proof, but rather to develop the skills to verbally express to their fellow students a convincing argument. In the process several mathematical ideas arise naturally and can be explored as part of the lesson plan.

The first mathematical idea is that of an invariant. The parity of the number of upside-down glasses is invariant in that this parity is preserved regardless of which adjacent pair of glasses is flipped. This observation is relatively easy to justify (we leave the justification to the reader) and leads directly to the second mathematical idea, induction. The invariant enables an inductive argument to be established to justify that the parity of the number of upside-down glasses determines the solvability of the puzzle. Again, we are not looking for a formal inductive proof, but rather informal yet convincing arguments based on induction (and we invite the reader to try this).

4. Summary and Conclusions

The two case studies discussed in this paper illustrate two quite distinct uses for the interactive whiteboard. The first, in Section 2, is more traditional in that it takes a core concept and uses the whiteboard to illustrate and explore this concept, all be it in a way that actively involves teacher and students. The use of the interactive whiteboard in this case enables the investigation of issues not easily explored with the traditional chalk board (e.g. how the motion of y is related to that of x).

The second case study, in Section 3, is less traditional as it uses the computer to set the scene so that deeper mathematical issues such as that of invariance and induction can be introduced. This case study requires complete involvement by the students. It not only encourages the development of mathematical thinking, but also the development of skills in verbalising reasoned arguments so as to convince others.

One of the important outcomes from our case studies is the confirmation of the vital importance of developing the software and lesson plan together, and for this development to involve the software engineer and the teacher working in partnership. We see this as essential if teachers are to use the interactive whiteboard effectively. Some commercial software vendors encourage the development of lesson plans that make use of their software packages (see for example [1]). However, the software package is fixed and it is the lesson plan that has to fit in with the software, rather than the two being developed together.

Another outcome from our work is the realisation that software needs to constantly evolve in order to meet the needs of individual teachers. A drawback with off-the-shelf software packages is that evolution usually results in extra features being added to the package but nothing being removed, with the consequence that the package becomes more complicated with each new release. By involving the software engineer in both software and lesson development it becomes possible to keep the software simple. This has benefits not just for the teacher, but also for the students who don't need to master a complex package in order to participate in the lesson. By customising software by working from underlying software code patterns and structures, it is relatively straightforward to add new features as required and remove deprecated features. The ideal is a separate software package for each concept.

A problem with customising software individually is that it requires a resource of software engineers. Clearly individual schools would find it prohibitive to directly employ such a resource. One way of easing the financial burden would be to establish a software support network to be shared between schools. This support network could take on responsibility for the ICT training of mathematics teachers through regular workshops, for giving advice about and demonstrating existing software, and for customising software as required.

The case studies in this paper demonstrate that effective mathematics-oriented software, fully integrated with the lesson plan, can be developed for the interactive whiteboard. By creating these packages as part of a working relationship between teacher and software engineer, the teacher is able to make full use of the technology. The resulting packages encourage the development of an interactive working partnership between teacher and students.

Acknowledgements Particular thanks go to John Mason, Sue Johnston-Wilder, Alan Graham and all at the Centre for Mathematics Education at the Open University. The case studies presented in this paper were developed while the first-named author was visiting the Centre in the Spring of 2004.

References

[1] Autograph [program]. Eastmond Publishing Ltd, 2004. (www.autograph-maths.com)

[2] Cabri Geometry II Plus [program]. CabriLog, 2004. (www.cabrilog.com/en)

[3] Cabri Jr [program]. CabriLog, 2004. (www.cabrilog.com/en)

[4] ClassPad300 [program]. Casio Computer co. Ltd., 2003. (www.casio.com/edu_e/)

[5] Curriculum Online. United Kingdom Department for Education and Skills. (www.curriculumonline.gov.uk)

[6] Derive 6 [program]. Texas Instruments, 2003. (education.ti.com/us/product/software/derive/features/features.html)

[7] Dwyer D., Ringstaf C. and Sandholtz J. Changes in teachers' beliefs and practices in technology-rich classrooms. *Educational Leadership*, **48**(8), pp. 45-52, 1991.

[8] Dwyer D. Apple classrooms of tomorrow: what we've learned. *Educational Leadership*, **51**(7), pp. 4-10, 1994.

[9] Fisher, C., Dwyer, D. and Yocam, K. *Education and Technology: Reflections on Computing in Classrooms*. Jossey Bass, San Francisco, 1996.

[10] Forgasz, H. and Prince, N. Computers for secondary mathematics: who uses them and how? In *Proceedings of the Australian Association for Research in Education (AARE)*, pp. 386-393, Fremantle, Australia, 2001.

[11] The Geometer's Sketchpad 4 [program]. Key Curriculum Press, 2003. (`www.keypress.com/sketchpad`)

[12] Goldberg B. and Richards J. Leveraging technology reform: changing schools and communities into learning organizations. *Educational Technology*, **35**(5), pp. 5-16, 1995.

[13] Greiffenhagen, C. From traditional blackboards to interactive whiteboards: a pilot study to inform technology design. In Nakahara and Koyama (eds), *Proceedings of the 24th International Conference of the Psychology of Mathematics Education Group*, Hiroshima University, **2**, pp. 305-312, Japan, 2000.

[14] Greiffenhagen, C. Interactive whiteboards in mathematics education: possibilities and dangers. In *WG11 The use of Technology in Mathematics Education*, ICME-9, Tokio/Makuhari, Japan, 2000.

[15] Hardley, M. and Sheingold, K. Commonalities and destinctive patterns in teachers' integration of computers. *American Journal of Education*, **101**, pp. 261-315, 1993.

[16] Mason, J., Burton, L. and Stacey, K. *Thinking Mathematically*. Addison-Wesley, 1982.

[17] Mathematica 5 [program]. Wolfram Research Inc., 2003. (`www.wolfram.com/products/mathematica/index.html`)

[18] Muir-Herzig R. Technology and its impact in the classroom. *Computers and Education*, **42**(2), pp. 111-131, 2004.

[19] Norton, S. Secondary mathematics teachers' responses to computers and their beliefs about the role of computers in their teaching. In *Making the difference: Proceedings of the Twenty-second Annual Conference of the Mathematics Education Research Group of Australasia (MERGA)*, pp. 404-410, Australia, 1999.

[20] Norton S., Campbell J. and Cooper T. Exploring secondary mathematics teachers' reasons for not using computers in their teaching: five case studies. *Journal of Research on Computing in Education*, **33**(1), pp. 87-109, 2000.

[21] Norton S. and Cooper T. Factors influencing computer use in mathematics teaching in secondary schools. In *Proceedings of the Mathematics Education Research Group of Australasia (MERGA) - Numeracy and Beyond*, Australia, 2001.

[22] Ofsted (Office for standards in education). *ICT in schools: effect of government initiatives on secondary mathematics*, report HMI 705, 2002. (`www.ofsted.gov.uk/public/docs02/ictsubject/maths_ictsec.pdf`)

[23] Perks, P. The interactive whiteboard: implications for software design and use. In Goodchild (ed) *Proceedings of the British Society for Research into Learning Mathematics*, **22**(2) pp. 55-60, 2002.

[24] Pollard, J. and Duke, R. A software design process to facilitate the teaching of mathematics. In *International Conference on Computers in Education (ICCE)*, pp. 906-907, Auckland, IEEE Computer Society Press, 2002.

[25] Pollard, J. and Duke, R. Creating a teacher and software-developer partnership. In *New Technologies and their Applications in Education: 6th International Conference on Computer Based Learning in Science (CBLIS)*, pp. 313-322, Cyprus, 2003.

[26] Pollard, J. and Duke, R. Revelations in the design of educational mathematical software. In 8th *Asian Technology Conference in Mathematics (ATCM)*, Taiwan, ATCM Inc, 2003.

[27] Polya, G. *How to Solve It*. Oxford University Press, 1957.
[28] Queensland Department of Education. *ICTs for Learning: School Information Kit 2003-2004*. Brisbane, 2003. (www.education.qld.gov.au/ictsforlearning)
[29] Sandholtz J., Ringstaff C. and Dwyer D. *Teaching with Technology: Creating Student-centered Classrooms*. Teachers College Press, New York, 1997.
[30] TI-89 and Voyage 200 [program]. Texas Instruments Inc., 2003. (education.ti.com/us/product/tech/89/features/features.html)
[31] TTA (Teacher training agency). *ICT and Mathematics: a Guide to Teaching and Learning Mathematics 11-19*, London, TTA/Mathematical Association, 2002. (www.m-a.org.uk/education/teachers_teaching_with_technology)