The Shaping of Student Knowledge: learning with dynamic geometry software

Keith Jones
University of Southampton, UK
dkj@soton.ac.uk

The focus of this paper is a software genre usually referred to as 'dynamic geometry' because of the ability of the user to dynamically manipulate geometrical figures created with the software tool. Using data from a longitudinal study of 12-13 students' use of dynamic geometry software, the focus of the analysis is on the interpretations the students make of geometrical objects and relationships when using this form of software. The analysis suggests that the students' mathematical reasoning is shaped by their interactions with the software in that their ability to explain geometrical facts and relationships evolves from imprecise, 'everyday' expressions, through reasoning that is overtly mediated by the software environment, to mathematical explanations of the geometric situation that transcend the particular tool being used. Such findings suggest that curriculum initiatives that encourage the use of dynamic geometry software are appropriate but that the incorporation of such software into classroom practices is unlikely to be straightforward.

Keywords: applications in subject areas; improving classroom teaching; interactive learning environments; pedagogical issues; secondary education

Introduction

In March 2005, the UK Department for Education and Skills published an ambitious "e-strategy" entitled *Harnessing Technology: transforming learning and children's services* (DfES, 2005). While not arguing for a "complete switch to new technology" (*ibid* para 25), the strategy paper is firmly of the view that "because ICT is an interactive medium, it is *ideal* for helping learners develop the skills they need …" (*ibid* para 85, emphasis added). Such a stance can appear to draw on the same technocratic assumptions as do previous (and, in some places, ongoing) systemic efforts at educational reform. Examples derived from these sorts of assumptions are the production of what might be termed 'teacher-proof' curriculum packages, ones that can be 'delivered' by any teacher in any classroom. In the case of the new technologies, the assumption appears to be that the character of their 'use' in teaching and learning is relatively invariant, transparent and unproblematic.

What this paper seeks to do is to go beyond these (often unspoken) assumptions about information and communications technology (ICT) by keeping the focus on the *potential* of ICT, cognisant that Cuban's (2001) critique of ICT as 'oversold and underused' continues to resonate with teachers. Rather than seeing ICT as 'ideal', the approach in this paper is to look to uncover some of the complexities of using ICT in the classroom as this is more likely to inform realistic policy decisions as well as inform future development of ICT applications (for examples and elaborations of these complexities, see, for instance, Artigue, 2000; Hennessy, Ruthven, & Brindley, 2005; Hoyles, Noss & Kent, 2005). In seeking to add to the evidence of the (sometimes underestimated) complexities involved in using ICT in teaching and learning, the perspective taken in this paper is that learning is always both supported and constrained by its context, a viewpoint that is reinforced by a considerable body of evidence. When this context involves the use of new technologies, some of this evidence comes from research that has at its root the assumption that human learning is fundamentally socio-cultural (for relevant work see papers ranging, for example, from Pea 1987, 1993, to Sutherland, 2004). From this point of view, learning is always 'learning to do something' with whatever 'cultural tools' are available, be they physical, intellectual and/or theoretical.

In this paper the object of analysis is the use of a software genre usually referred to as 'dynamic geometry' due to the ability of the user to dynamically manipulate geometrical figures created with the software tool. From a socio-cultural perspective, such a 'cultural tool' is very interesting because its design attempts to capture a version of that element of human knowledge that is plane geometry - something that renowned mathematician Sir Michael Atiyah (2001) consider as one of the two pillars of mathematics (the other being algebra).

A pivotal question is how we might distinguish the cognitive tools and cultural artefacts that are likely to contribute to providing learners with the capability of becoming what Elliott (2000: 139) refers to as 'autonomous and free agents capable of shaping the conditions of their existence in civil society'. Is dynamic geometry software such a tool? What are its affordances and how might we determine its constraints on learning? This is the focus of this paper.

In what follows, the paper begins with a brief account of the issues involved in students learning to use ICT applications. This leads to a short review of the existing research on the use of the dynamic geometry software tool. The main part of the paper reports selected findings from a longitudinal study of 12-13-year-old students' use of dynamic geometry software (for more details, see Jones, 2000). The analysis focuses on the possible shaping of the students' mathematical reasoning through their interactions with the software and places this analysis in the context of curriculum initiatives, frequently at a national level, that encourage the use of dynamic geometry software.

Students learning with and through ICT

A range of research has shown that, despite gaining experience of an ICT application through continued use, many users with basic knowledge of a software application do not necessarily progress beyond techniques they learn in the early stages and do not always go on to make the most efficient and effective use of the application (see, for example, Bhavnani & John, 1997; Nilsen, 1993). Such existing studies have generally focussed on the strategies that might allow users to learn the efficient use of the software. Yet, as Selinger (2001) explains, there can be a tension between teaching *about* the ICT tool and teaching *through* the use of an ICT tool. In this paper, as the aim is to ensure that the application is a tool that supports and enhances learning, the focus is on how students might be enabled to make the tool work for them. In the classroom this may not happen straightforwardly. For example, John and Sutherland (2004: 107) found that classroom learning with an ICT tool "might breed more confusion before genuine understanding occurs".

Research on the use of the dynamic geometry software (DGS)

Dynamic geometry software (DGS) provides a variety of tools for constructing geometric objects using a range of `primitive' objects (such as points, segments, lines, circles etc.). The tools available in the software include 'classical' geometrical constructions (midpoint, perpendicular, parallel, etc.) as well as transformations (reflect, rotate, translate, etc.) Once drawn, measurements can be taken (length, angle, area, etc.). The 'dynamic' aspect of the software comes from the ability to drag the `primitive' objects, such as points and lines, around the screen with the mouse (or tracker-ball on a laptop, or even the finger when using an interactive whiteboard). While such dragging deforms the geometrical figure that has been constructed, some aspects remain the same - depending on the geometrical ideas used in the construction. In this way, the software allows a focus on the key geometrical idea of invariance and provides means for users to explore geometrical theorems (in addition, forms of DGS usually includes the means for drawing loci, performing animations, and working with coordinates, thus permitting a wide range of geometrical activity).

The development of this form of software has sparked considerable interest in the mathematics education community, not only because of the central role of geometry in mathematics, but also because providing a meaningful experience of deductive reasoning and proof for students at the school level appears to be difficult (for recent reviews of student difficulties with proof, see Hanna and Jahnke, 1996, Dreyfus, 1999). A variety of DGS research reviewed by Jones (2002) shows that interacting with DGS can help students to explore, conjecture, construct and explain geometrical relationships. It can even provide them with the basis from which to build deductive proofs. Overall, this research has found that classroom discussion and group work continue as important components of effective teaching and learning. The research suggests that DGS cannot provide a "self-contained" environment for learning geometry, but that other activities are needed for students to make progress in this area of mathematics. In particular, the teacher plays a crucial role in guiding students to theoretical thinking in geometry.

Much existing research with dynamic geometry software (DGS) has quite properly focused on students in upper secondary (senior high) school, especially in situations where the students have received considerable teaching input in plane geometry, including the proving of elementary theorems, but where they may be new to the particular software. What is less clear at the moment is what impact the use of dynamic geometry software has on students in lower secondary (junior high) school where students have limited experience of the formal aspects of geometry but where contact with geometrical theory through the software may be especially valuable in providing a foundation for further work on developing deductive reasoning in mathematics. Research is especially needed on the nature of the tasks that students can tackle using DGS, the form of teacher input (and its impact) and the role of the classroom environment and culture (expectations, working methods, etc) as ll these things impact on the form of knowledge students gain from learning geometry with DGS.

Research aim, design and methods

The aim of the research reported in this paper was to identify how lower secondary (junior high) school students reason about geometrical objects and relations as they experience them through the dynamic geometry environment and how the mathematical explanations they offer evolve as they become more experienced both with geometry and with the software. The empirical work was designed to be carried out in the UK and, following Hoyles (1997), was informed by the structure of the mathematics curriculum experienced by students in the UK. As Hoyles describes, while *formal* mathematical proof is likely to be restricted in the UK to the most able students only (and mostly encountered by students in upper secondary school), the curriculum does provide for opportunities for conjecturing and presenting generalisations at all levels. Wherever possible, design choices were made with a view to the typicality of the setting. The school selected for the empirical work was an urban comprehensive school whose results in mathematics at age 16 were at the national average (there is a national system of testing in the UK that allows such judgements to be made). The chosen geometrical topic, the classification of quadrilaterals, is a standard component of the geometry curriculum for the selected age group of students, both in the UK and around the world (for related work, see Fujita & Jones, 2003).

In line with national UK expectations (although these are not always borne out in practice in UK schools), the mathematics teachers in the school customarily used a problem-based approach to teaching mathematics and the students often worked in pairs or small groups on mathematical problems and regularly used computers. Throughout their mathematics work, the students were expected to be able to explain the mathematics they were doing, either orally or in writing. This meant that work on geometry using dynamic geometry software would fit with the usual experience of the students. The classes of 12-13 year-olds in the school had four 50-minute mathematics lessons per week.

The particular class of 12-13 year-olds selected for the research were judged typical of that suitable for studying the chosen geometrical topic in that they were above-average in mathematics for their age (the school allocated students to different mathematics classes according to attainment in mathematics tests). A longitudinal design was prepared, utilising a series of teaching units that were developed in collaboration with the regular teacher of the class. These teaching units addressed the selected geometrical topic and could be accommodated in the regular routine of the class. For most of the 9 months of the study, up to four computers were available in the classroom. This meant that, as pairs of students took turns in using the computers, there might be gaps of up to a week between sessions that any particular pair of students had using the software. During these ‘gaps’ the students undertook other mathematics work, including some geometry topics involving area and volume, but none were directly concerned with the geometric topic being studied in the project.

During the project, the following data were collected: video and additional audio tape to capture the onscreen work and student-student and student-teacher interactions, student written work (unaided), student software files, the ‘history’ of the student constructions using the software (a feature available with the particular software), and researcher field-notes. All the students in the class were tested using an appropriate geometry test prior to the start of the unit of work and on its completion. The DGS used was *Cabri I* for the PC.

Findings and discussion

The data selected for discussion are taken from each of the teaching units devised for the project.

The first teaching unit served to introduce the students to the software and to the constraint of robustness of a figure under drag. Analysis of the data from this phase of the project raised several issues about the way the students were interpretating the features of the software environment. Some of these were fairly trivial. For example, the students had to realise the distinction between lines (that are infinite) and line segments – a distinction that these particular students had not met previously. Other aspects of the software environment took longer for the students to become accustomed to. The most significant of these, in terms of the impact on the students learning, were that certain objects (so-called basic objects) can be dragged while others (constructed using these basic objects) cannot, that certain points (notably points of intersection) have distinctive properties (for example, points of intersection cannot be dragged), and that there can be a sequential organisation required in order to construct a figure that is "robust" under drag (ie for which the relevant geometrical properties remain invariant) – for more on dragging, see, for example, Hölzl, 1996).

The nature of points of intersection between lines was especially perplexing for some of the students. For example, early on in using the software, one student asked:

> *What's the intersection doing? Does it keep the dot there?*

While, later in that lesson, the other student in the pair explains:

> *You have to make an intersection between those two lines so that they can't be moved.*

This (incorrect) interpretation of points of intersection has also been observed by Ainley and Pratt, 1995)

In the second teaching unit the students worked through a series of three tasks that involved constructing the following quadrilaterals: a rhombus, a square, and a kite. The third, and final, teaching unit involved the students working through a series of six tasks that related to the relationships between various quadrilaterals, culminating in a task which asked the students to complete a hierarchical classification of the 'family' of quadrilaterals and explain the relationships between the various quadrilaterals.

During these latter teaching units, the students continued to struggle with learning the mathematics involved *and* with coming to terms with the nature of the software. In the first task of the second teaching unit, for example, the students were asked to construct a rhombus and explain why the shape is a rhombus. A typical explanation from a pair of students is as follows:

Pair A (written explanation):

> *The radius is the same for the circle and the diamond [the rhombus] and we made the diamond from the help of the first construction. The sides are all the same because if the centre is in the right place, the sides are bound to be the*

> *same. The diagonals of the diamond cross in the middle though they are different size (length). They cross at the middle through the line.*

Here there is evidence of a lack of student capability with precise mathematical terminology (viz. the use of term ‘diamond’ rather than rhombus, even though they knew the mathematical term) and some reliance on perception rather than mathematical reasoning (for example, “it looks straight”). As the students are familiar with the properties of a rhombus they are able to list these. However, they are unfamiliar with the notion that ‘formal’ mathematical definitions contain only necessary and sufficient properties, and with how the properties of geometrical figures are inter-related (beyond writing, for example, that “if the centre is in the right place, the sides are bound to be the same”).

As the teaching units progressed, the students ideas became more mathematically sophisticated. In the first task of the third teaching unit, for example, the students were set the challenge of constructing a rectangle that could be modified to a square. The students were then expected to explain why all squares are rectangles (utilising what mathematicians refer to as an ‘inclusive definition of a rectangle, see de Villiers, 1994).

One pair wrote:

> *You can make a rectangle into a square by dragging one side shorter and so the others become longer until the sides become equal.*

This is a reasonable explanation of how a rectangle can ‘become a square’. What is interesting is that the students’ explanation is couched in terms of the nature of the software environment, including the use of the term “dragging” and about something ‘becoming’ something else.

In a later task, the students are asked to construct a trapezium that can be modified to a parallelogram and thereby explain why all parallelograms are trapeziums. These are the written explanations of two of the pairs of pupils:

Pair A (written explanation):

> *It is a trapezium because it has one pair of parallel lines. A parallelogram is parallel both ways.*

Pair C (written explanation):

> *Trapeziums have one set of parallel lines and parallelograms have two sets of parallel lines.*

This is evidence that, by this stage, the students' explanations are not couched in terms related to the nature of the software environment. The cause of this shift was mainly due to the role of the teacher in consistently referring to the geometrical properties of the shapes whenever there was an interaction with the students (the role played by the teacher, and the impact of the teacher-student interactions, is the subject of continuing separate analysis).

Conclusion

This paper reports selected findings from a longitudinal study of 12-13 students' use of dynamic geometry software. The analysis, which focused on the interpretations the students make of geometrical objects and relationships created using this form of software, suggests that the students' mathematical reasoning is shaped by their interactions with the software in that their explanations evolve from imprecise, 'everyday' expressions, through reasoning that is overtly mediated by the software environment, to mathematical explanations of the geometric situation that transcend the particular tool being used.

Such findings suggest that the current curriculum initiatives that are encouraging the use of dynamic geometry software in secondary school mathematics classrooms are appropriate but that the incorporation of such software into classroom practices is unlikely to be straightforward. With small amounts of learner use, there may be no impact, while in the medium term, learner knowledge may be heavily influenced by features of the software that have nothing to do with geometry but may be more related to the reification of geometry through the realisation of the software environment.

Evidence from this study, along with other studies of DGS reviewed by Jones (2002), does suggest that this form of software has the potential to contribute to providing learners with the capability of which Elliott (2000: 139) speaks. Using such software does afford access to geometrical theory in a way that allows the learner to experience the problem solving choices of what Elliott refers to as an "autonomous and free agent". Yet the constraints on learning that occur through the mediational impact of the software, especially in the early to middle stages of use, are such that the complexity of students learning geometry with this form of software tool should not be underestimated.

Acknowledgements

Thank go to the staff and students of the school where the empirical study was carried out, especially the teacher who helped with the design of the classroom tasks. Aspects of this research were part-supported by grant A94/16 from the University of Southampton Research Fund.

References

Ainley, J. and Pratt, D. (1995). Mechanical Glue? Children's perceptions of geometrical construction. *Proceedings of the 3rd British Congress on Mathematical Education*. Manchester. Part I, p 97-104

Artigue, M. (2000). Instrumentation issues and the integration of computer technologies into secondary mathematics teaching. *Developments in mathematics education in German-speaking countries: selected papers from the annual conference on didactics of mathematics, Potsdam, 2000*. Gesellschaft für Didaktik der Mathematik (GDM).

Atiyah, M. (2001). Mathematics in the 20th century: geometry versus algebra. *Mathematics Today*, *37*(2): 46–53.

Bhavnani, S. K., & John, B. E. (1997). From sufficient to efficient usage: an analysis of strategic knowledge. In: S. Pemberton (Ed.), *Looking to the future: proceedings of the ACM CHI'97 conference*. New York, NY: Association for Computing Machinery.

Cuban, L. (2001). *Oversold and underused: computers in the classroom*. Cambridge, MA: Harvard University Press.

Department for Education and Skills (2005). *Harnessing technology: transforming learning and children's services*. London: DfES.

de Villiers, M. (1994). The role and function of a hierarchical classification of quadrilaterals. *For the Learning of Mathematics*. *14*(1), 11-18.

Dreyfus, T. (1999). Why Johnny can't prove. *Educational Studies in Mathematics*, *38*(1-3), 85-109.

Elliott, J. (2000). Towards a synoptic vision of educational change in advanced industrial societies. In: H. Altrichter & J. Elliott (Eds), *Images of EDUCATIONAL CHANGE*. Milton Keynes: Open University Press.

Fujita, T. and Jones, K. (2003). Interpretations of National Curricula: the case of geometry in Japan and the UK. Paper presented at the *British Educational Research Association Annual Conference*, Heriot-Watt University, 10-13 September, 2003.

Hanna, G. and Jahnke, H. N. (1996). Proof and proving. In: A. J. Bishop (ed), *International handbook on mathematics education*. Dordrecht: Kluwer.

Hennessy, S., Ruthven, K., & Brindley, S. (2005). Teacher perspectives on integrating ICT into subject teaching: commitment, constraints, caution and change. *Journal of Curriculum Studies*, *37*(2) 155-192.

Hölzl, R (1996). How does 'dragging' affect the learning of geometry. *International Journal of Computers for Mathematical Learning*. *1*(2) 169-187.

Hoyles, C. (1997). The curricular shaping of students' approaches to proof. *For the Learning of Mathematics*, *17*(1), 7-16.

Hoyles, C., Noss, R. & Kent, P. (2005). On the integration of digital technologies into mathematics classrooms. *International Journal of Computers for Mathematical Learning*, *9*(3), 309 - 326.

John, P. & Sutherland, R. (2004). Teaching and Learning with ICT, New Technology, New Pedagogy? *Education, Communication and Information Journal (ECi)*, *4*(1), 101-107.

Jones, K. (2000). Providing a foundation for deductive reasoning: students' interpretations when using dynamic geometry software and their evolving mathematical explanations. *Educational Studies in Mathematics*, *44*(1&2), 55-85.

Jones, K. (2002). Research on the use of dynamic geometry software: implications for the classroom. *MicroMath*, *18*(3), 18-20 & 44-45.

Nilsen, E., Jong, H., Olson, J. S., Biolsi, K., Rueter, H. & Mutter, S. (1993). The growth of software skill: a longitudinal look at learning & performance. In: Ashlund, Stacey, Mullet, Kevin, Henderson, Austin, Hollnagel, Erik, White, Ted (eds.): *Proceedings of the ACM CHI 93 Human Factors in Computing Systems Conference*. April 24-29, 1993, Amsterdam, The Netherlands. p.149-156

Pea, R. D. (1987). Cognitive technologies for mathematics education. In: A. H. Schoenfeld (ed), *Cognitive science and mathematics education*. Hillsdale, N.J.: Lawrence Erlbaum Associates, pp89-122.

Pea, R. D. (1993). Practices of distributed intelligence and designs for education. In: G. Salomon (ed), *Distributed cognitions: psychological and educational considerations*. Cambridge: CUP, pp47-87.

Selinger, M. (2001). Learning information and communications technology skills and the subject context of the learning. *Journal of Information Technology for Teacher Education*, *10*(1&2), 143-156

Sutherland, R. (2004). Designs for learning: ICT and knowledge in the classroom. *Computers and Education*, *43*(1-2), 5-16.