CHAPTER 12

# KEEPING THE FAITH

## Fidelity in Technological Tools for Mathematics Education

**Thomas P. Dick**

There are many different audiences that can benefit from the insights that can arise from thoughtful research on the use of technology in the teaching and learning of mathematics. These audiences include classroom mathematics teachers, mathematics curriculum developers, educational policymakers, and the designers of the technological tools. The purpose of this chapter is to consider, from the perspective of the tool designer, the lessons that can be learned from research.

As the title indicates, the theme of this chapter is *fidelity*—technology designed for use in mathematics education should be faithful to some basic principles. We will refer to these principles as *pedagogical fidelity*, *mathematical fidelity*, and *cognitive fidelity*.

Before providing specifics of what we mean by "keeping the faith" pedagogically, mathematically, and cognitively, it is important to note that these remarks are aimed at designers of technological tools intended primarily to facilitate the *learning* of mathematics. There are, of course, tools

*Research on Technology and the Teaching and Learning of Mathematics: Vol. 2. Cases and Perspectives*, 333–339

that are designed primarily for the *application* of mathematics that enjoy extensive use in mathematics classrooms. For example, a computer spreadsheet is a tool that has been heavily adapted for use in the mathematics classroom, in ways that likely were never imagined by the designers of the first spreadsheets. For a tool such as a spreadsheet, the lessons to be learned from research are actually intended for the educational adapters (classroom teachers, curriculum developers, etc.), not the spreadsheet designers.

We will not go so far as to say that our distinction between designed and adapted tools is universally easy to apply. For example, the suite of features found on some computer algebra systems (CASs) could be argued to be a mix of tools for mathematics learning and mathematics applications.

## PEDAGOGICAL FIDELITY

At the risk of invoking an educational cliché, we take as axiomatic that students learn mathematics by doing mathematics. But what does it mean to "do mathematics?" Suppose we make our axiom a bit more explicit.

> *Axiom:* Students learn mathematics by taking mathematical actions (e.g., transforming, representing, manipulating) on mathematical objects (e.g., symbolic expressions, graphs, geometric figures, physical models), observing the mathematical consequences of those actions, and reflecting on their meanings.

Students' reflections on mathematical consequences of mathematical actions on mathematical objects are the fuel for feeding the cycle of prediction-conjecture-testing that ultimately leads to proofs or refutations.

A technological tool stays true to this pedagogical axiom to the degree that its use is perceived transparently by the student to further these purposes. That is, for a tool to be pedagogically faithful, the student should perceive the tool as (a) facilitating the creation of mathematical objects, (b) allowing mathematical actions on those objects, and (c) providing clear evidence of the consequences of those actions.

Keeping the pedagogical faith is evidenced most clearly in the organization of the user interface of a technological tool. For example, consider something as mundane as the menu organization of a software package. Clearly the software will have general commands common to almost all packages (file management, editing functions such as cut/copy/paste, etc.). We would also expect the software to have commands that correspond to creating mathematical objects, taking actions on those objects, and displaying or reporting information. There is no single best way to



organize this mix, but if the designer stays faithful to the pedagogy principle, then that organization should be made with the clear goal of facilitating the mathematical moves with as few distractions as possible. For example, a graphing software package should not mix parameter controls that are cosmetic with those having mathematical substance. Font characteristics for labels or measurements are user preferences that should be quite separate from window scaling options.

The pedagogical faithfulness of a technological tool can be reflected in how a user would typically describe the steps in an activity or procedure. A pedagogically faithful tool will lend itself to describing moves in terms of interactions with the mathematics (e.g., "I graphed this function," "I created this triangle," or "I measured this area") rather than in terms of interactions with the tool (e.g., "I went to this menu," "I changed this mode," or "I set the preferences to").

## MATHEMATICAL FIDELITY

A technological tool must stay true to the mathematics. If a student perceives that a virtual mathematical object has been created through the use of a tool, then the characteristics and behavior of that object in the technological arena should reflect accurately the mathematical characteristics and behavior that the idealized object should have. (Platonism is not a bad philosophy for the technological tool designer!)

While the goal of mathematical fidelity may seem obvious, in practice it can be quite difficult to implement. Some of the obstacles are inherent and due to technological limitations. Other problems are due to conscious design choices that put ease of use at a higher priority than faithfulness to mathematical structure.

For example, design choices made in the handling of implicit multiplication in algebraic expressions may make life easier for the user in certain circumstances, but betray mathematical conventions in other circumstances. To allow users to enter an expression such as sin $2x$ as the conventional shorthand for sin($2x$) means that implicit multiplication must have a higher precedence than function application. This convenience puts us in a bizarre situation in which explicit multiplication notation has an entirely different mathematical interpretation from implicit multiplication: sin $2*x$ would mean (sin(2))$*x$, since the evaluation of the sine function (at 2) would take precedence over the explicit multiplication by $x$.

A common technological limitation has to do with the modeling of continuous phenomena with discrete structures. This goes far beyond the usual caveats of interpreting displayed numerical results represented as finite precision terminating decimals. The hidden traps are encountered

when the results of such computations are generated and used but are not seen by the student.

For example, suppose a geometry tool that provides features for the creation and manipulation of lines in the plane uses the Cartesian slope of the line as one of the underlying defining parameters. That is, the tool stores a line internally using two pieces of data: the slope of the line and a reference point (such as the $y$-intercept). Of course, the case of a vertical line would need to be handled differently by such a tool. The tool could use this very compact bundle of data not only to generate a visual display of the line in a given window but also to perform actions such as creating a parallel or perpendicular line to a given line or testing for the parallelism or perpendicularity of two given lines. The real problems arise when dealing with lines that are *nearly* vertical or horizontal. Precision limitations could result in gross (and visually obvious) errors. Indeed, my first experience attempting to create a perpendicular to a very steep (visually almost vertical) line on an early dynamic geometry package resulted in a line that was nowhere near horizontal visually.

Perhaps the worst "conscious" offenses have been made with how technological tools deal with functions. The mathematical concept of function is arguably the single most important in all of mathematics, so this lack of faithfulness is particularly troubling. In most cases, the fundamental error made with functions is to treat them as expressions rather than mappings with *domains*. The mathematics community itself is somewhat to blame for the situation, given the common convention of adopting implicit domains for functions in certain contexts. For example, it is typical in calculus to assume that a functional expression $y = f(x)$ defines a real-valued mapping having as its domain the largest subset of real numbers $x$ for which the expression $f(x)$ results in a real number. Thus, one can "define" a function simply by writing $y = \frac{1}{x-2}$, provided it is understood that the implied domain of the function is $\{x : x\varepsilon\Re, x \neq 2\}$.

We are already on shaky ground pedagogically with such conventions, for while the expert may be careful to account for these implicit domains in composing functions, a student may be blissfully unaware of them. The risk is compounded when a student uses a technological tool that "defines" functions without reference to any domain. For example, the equations $y = 2\ln(x)$ and $y = \ln(x^2)$ would define different functions having different implicit domains using the usual calculus convention. The domain of the first is the set of positive real numbers and the domain of the second is the set of nonzero real numbers, although it is true that the two functions take on the same values at each point in the intersection of their domains.



Now consider the function $y = |\ln(x)|$. What is its domain? If we follow the chain of compositions, $x \rightarrow \ln(x) \rightarrow |\ln(x)|$, the implicit domain would be the set of positive real numbers. However, many computer and calculator graphers will plot this function as if its domain were the set of all nonzero real numbers, since the *modulus (absolute value)* of the complex number $\ln(x)$ that results for $x < 0$ gives a final result of a real number.

The algebraic "simplification" of an expression can effectively change the domain (and hence change the definition of the function). Consider the function $y = arc\tan\left(\frac{1}{x}\right)$, implicitly defined on the domain of all nonzero real numbers. Its derivative is given by $\frac{dy}{dx} = \frac{1}{1+\left(\frac{1}{x}\right)^2} \cdot \frac{-1}{x^2} = -\frac{1}{x^2+1}$.

The final simplification of the algebraic expression for the derivative function results in an expression whose implied domain is the set of *all* real numbers, including $x = 0$. Clearly, the derivative of a function cannot be defined at a point that is not in the domain of the function! At this point, we must make the domain of nonzero real numbers *explicit* in order to maintain the mathematical fidelity of our actions (differentiation followed by simplification). This particular example is sometimes used as a test of the integrity of computer algebra systems.

It is unrealistic to expect the mathematics community to suddenly drop conventions that have the inertia of historical usage behind them. Human users of symbols must be aware of the implicit assumptions and contexts for their usage (Is $f(x) = 3$ an equation or a definition of a constant function?) and ignore them at their own risk. To use technological tools intelligently in arenas in which implicit assumptions and contexts lurk, there need to be ways of explicitly informing the machine. One strategy has been the setting of modes or preferences that correspond to these explicit assumptions. However, for tools that aim to handle many different possible mathematical contexts, the number of mode/preference settings can quickly become unwieldy.

## COGNITIVE FIDELITY

Technological tools for mathematics education should be faithful to students' cognitive processes. Unlike the application of a technological tool in the "real world," there should be more emphasis on illuminating mathematical thinking processes than simply arriving at "black box" final results as efficiently as possible.

In the context of intelligent tutoring systems, Beeson (1989) has used the term "cognitive fidelity" to refer to the degree to which the com-

puter's method of solution resembles a person's method of solution. In that same context, Beeson discusses the idea of a "glass box"—a computer program that allows the user to see *how* the answer is produced, meaning that a program is most useful in instruction if its (visible) inner workings are cognitively faithful. (We would note that this usage of the term "glass box" applied to a computer program is a bit different from the term "white box" used to describe instructional activity with a CAS (see, for example Cedillo & Kieran, 2003). The machine algorithms used by a tool to accomplish a result may bear little resemblance to conceptually based procedures a person typically would use. For example, the numeric root finder or equation solver found on most graphing calculators generally uses techniques far more sophisticated than those with which its users (students) would be familiar.

Some of the calculus reform projects funded by the National Science Foundation did not have to do so much with curriculum development as they did with the building of a glass box front end (i.e., an additional interface layer through which the user works instead of the usual interface) to a CAS. A CAS generally uses some very sophisticated "black box" symbolic integration techniques to produce antiderivatives. A classical integration by parts ($\int u dv = uv - \int v du$) completed "by hand" could be performed quite differently by a CAS. A glass box for integrating by parts might require the student to choose the $u$ and $dv$, calculate the corresponding $du$ and $v$ (with the help of the CAS, perhaps) and then instruct the CAS to perform the integration by parts using the student's choices.

## CHALLENGES FOR THE FUTURE: THE NEED FOR AUTHORING TOOLS

The greatest unrealized challenge is to bridge the gap between the vision of the educational practitioners and the technical expertise of the educational tool designer. Specifically, how do we efficiently move from the idea for a mathematics learning activity to the implementation of a technological tool that facilitates that activity and remains faithful to it? Currently, the vast majority of classroom teachers and curriculum developers find themselves adapting their ideas and activities to the available tools. In the process, the original ideas may be compromised. In some rare cases, the practitioner may have the technological savvy or programming knowledge (or sufficient communication access to those who do) to design or adapt the tool to the activity in a way that does not compromise the original vision.



What we need to move toward is the notion of authoring tools—that is, tools for building tools—that provide a kind of "macro" construction kit for practitioners to create their own learning microworlds for students, complete with user-friendly interfaces and robust operational integrity. We have seen some exciting possibilities for how technology can be used to aid mathematics learning and teaching that have emerged in the hands of a few developers who share both educational vision and technical expertise. Now think about a future in which the number of such developers is multiplied literally by the thousands. Authoring tools that could truly enable the classroom teacher or curriculum developer to implement their ideas directly into reliable ready-to-use technological tools for mathematics learning would shift our language from technological innovation to educational innovation.

In terms of the fidelity principles discussed in this chapter, the structure of the authoring tools themselves could help practitioners who are developers adhere to those principles.

A dream perhaps? Keep the faith!

## REFERENCES

Beeson, M. (1989). Logic and computation in MATHPERT: An expert system for learning mathematics. In E. Kaltofen & S. Watt (Eds.), *Computers and mathematics* (pp. 202-214). New York/Heidelberg: Springer-Verlag.

Cedillo, T., & Kieran, C. (2003). Initiating students into algebra with symbol-manipulatingcalculators. In J. T. Fey, A. Cuoco, C. Kieran, L. McMullin, & R. Zbiek (Eds.), *Computer algebra systems in secondary school mathematics education* (pp. 219-239). Reston, VA: National Council of Teachers of Mathematics.