A Proposal for the Development of an Interactive Mathematics Laboratory for Mathematics Education*

William M. Farmer

Department of Computing and Software
McMaster University
1280 Main Street West
Hamilton, Ontario L8S 4L7
Canada

wmfarmer@mcmaster.ca

12 June 2000

1 The Nature of Mathematics

Like Janus, mathematics has two faces. One face is a huge body of knowledge assembled over thousands of years. The other face is a process by which mathematical knowledge is obtained. Mathematics education has traditionally emphasized the first face of mathematics. Students are taught many mathematical facts, but they are rarely taught what the mathematics process is and how to employ it. As a consequence, engineers and scientists today often do not know how to *do* mathematics, and the population as a whole does not understand what mathematics is and does not appreciate its importance in today's world.

The mathematics process is both a creative and explorative process. Models about the mathematical content of the world are *created* by defining mathematical structures and concepts, and the models are *explored* by performing calculations and stating and proving conjectures. The process is the most important face of mathematics. It is the fountainhead from which mathematical knowledge flows and a central component of the infrastructure that makes today's technological and informational world possible.

Every mathematics student needs to learn the mathematics process. Of course, the best way to learn it is to practice it. But high school and university students are given little chance to experience the full process of creating and exploring mathematics. Instead, they are taught merely about byproducts of the process—facts and calculations separated from the process which made them. Many mathematics educators feel that trying to teach students to learn the mathematics process is unproductive, and accordingly, they have argued, for

* Presented at the Workshop on Deduction Systems for Mathematics Education, CADE-17, Carnegie Mellon University, Pittsburgh, Pennsylvania, June 16, 2000.

example, that mathematical proof should be deemphasized in, or even eliminated from, the standard mathematics curriculum.

It is true that most people find it difficult to do mathematics. All mathematics creation and exploration is performed in some context, which includes vocabulary and notation for expressing concepts and assertions, and assumptions and rules for governing calculation and proof. In ordinary (informal) mathematics, the context is almost entirely implicit and often substantial mathematical expertise and training is needed to "see" it. The average person is often not able to see it. In other words, most people, if they do mathematics at all, they do it blind.

2 Interactive Mathematics Laboratories

An *interactive mathematics laboratory* (IML) is a computer system with a set of integrated tools designed to facilitate the mathematics process. It is a *formal* environment where the mathematical context is fully explicit. It is an *interactive* environment where the user can create and explore mathematics. And it is a *mechanized* environment where the system can perform many functions useful to the user.

An IML would provide the following services to mathematics students as well as to engineers, scientists, and mathematicians:

1. **Context Management.** An IML would keep track of the context the user is working in. The definitions and assumptions in force would be explicit. The user would have the freedom to examine the contents of the context and to modify, extend, or switch the context as desired. The rules of rigorous reasoning would be effectively encoded in an IML, and the system would check the soundness of the operations performed by the user. Consequently, the "rules of the game" would be apparent to the user, and all attempts to make unfounded conclusions would be immediately identified.
2. **Mathematics Library.** An IML would have access to a large library of electronically stored mathematics organized as a network of *axiomatic theories*. The mathematical information in the library would be dynamically represented, and so requested information could be generated on the fly. It would include both algorithmic and axiomatic mathematics. The theories would be linked via *theory interpretations* [7, 23] which would serve as conduits through which information from one theory could be "transported" to another theory [2]. This would enable the library to offer multiple views of the same mathematics.
3. **Creation.** An IML would facilitate the creation of mathematical ideas and objects, including mathematical models, definitions, conjectures, proofs, programs, notation, axiomatic theories, and links between theories. The user could start from scratch or build on the work of others. The creations could be added to the theory library, and if desired, made available to the public. This facility could be used by students to learn how mathematics is created.
4. **Exploration.** An IML would provide the ability to "browse" the theory library in sophisticated ways. For example, the user could ask to see all

the different ways of defining a continuous function and then explore what impact these different definitions have on the basic development of calculus. A user could ask what a certain complex expression reduces to in some particular algebraic theory. After choosing a well-known theorem, a user could ask for other theorems in other theories that are generalizations of it. And a user could study a selected proof to whatever level of detail is desired.

5. **Mathematical Proof.** An IML would support the development of mathematical proofs. Proof is what separates mathematics from all other disciplines; it is the prime instrument of the creative/explorative process of mathematics. IML-produced proofs would be both humanly comprehensible like the informal proofs of mathematicians and mechanically manipulatable like the formal proofs of logicians. An IML would offer a new kind of proof that is easier to understand and to construct.
6. **Calculation.** Like a computer algebra system, an IML would be able to perform calculations using both symbolic and numeric computation. However, unlike a contemporary computer algebra system, an IML would perform its calculations within a rigorous logical framework and could perform context-directed calculations [12]. Moreover, many calculations could be performed in a "transparent" mode in which the steps of the calculation would be visible to the user.
7. **Automated Deduction.** Automated deduction would be utilized systematically in an IML, particularly to handled low-level details. This would allow the user to focus her reasoning on the key aspects of a problem and let the system handle the more routine aspects. The level of automated deduction could be controlled, which would be particularly important for students. For each student, an IML would establish a demarcation between what is routine and what is key. The system would largely handle what is routine, while the student, with only limited help from the system, would be responsible for what is key. The frontier between what is routine and what is key—and what is automated and what is not–would be moved forward as the student mastered new concepts and techniques.
8. **Organizational Support.** An IML would automatically handle the lion's share of the organizational concerns and the drudgework involved in doing mathematics. For example, mathematics often involves long, sometimes tedious, threads of reasoning that are very difficult to keep track of with only the help of pencil and paper and one's memory. Good organizational support provided by a machine would allow the user to concentrate more fully on the key concepts and techniques being studied or applied. It would also enable students to work on more realistic and better focused problems.
9. **Notational Freedom.** Since an IML manipulates and stores mathematics represented in a precise, formal manner, it would be easy to display the mathematics using multiple notations. The user could choose whatever notation she preferred, and mathematics developed by one user in one notation could be viewed by another user in another notation.

An IML which could offer these kinds of services would have the potential to transform how people do mathematics, and in particular, how students learn mathematics. An IML would greatly extend their mathematical reach, allowing them to learn more mathematics by being able to do more mathematics.

3 Related Technology

IMLs do not exist today. However, a substantial portion of the technology needed for building an IML can be found in contemporary mechanized mathematics systems. Computer algebra systems, such as Axiom [15], Macsyma [16], Maple [4], and Mathematica [24], offer a rich collection of techniques for performing symbolic computations. Theorem proving systems, such as Automath [19], Coq [1], EVES [6], HOL [14], IMPS [10, 11], Isabelle [20], Mizar [22], NQTHM [3], Nuprl [5], Otter [17], and PVS [21], have much of technology that an IML needs, but they are more narrow in scope than an IML and are very difficult to use without a fairly deep understanding of formal mathematics.

These systems are a significant step toward an IML. They demonstrate the feasibility of an IML and the impact an IML could have on mathematics practice. They also demonstrate that the usefulness of an IML will depend at least as much on good system design and user interface software as on good logical and deductive machinery.

4 Obstacles

The potential benefits of an IML are great, but the obstacles that stand in the way of developing an IML and making it part of mathematics practice are also great. The following are some of the major obstacles:

1. **Development Cost.** The cost of developing an effective IML that is accessible to ordinary students would be very high, on the order of several million U.S. dollars. A development team for an IML would require funding for an extended period of time and would need a wide range of expertise in areas including mathematics, logic, automated deduction, symbolic computation, human-computer interaction, and mathematics education.
2. **Mathematics Community.** The mathematics community has largely ignored the field of mechanized mathematics. Although many mathematicians use computer algebra systems, only a small fraction of mathematicians are actively involved in developing or applying theorem proving systems. Their knowledge and leadership, however, is essential for the development and deployment of an IML. They need to be in the vanguard, not on the sidelines.
3. **Accessibility.** Contemporary theorem proving systems are only accessible to highly sophisticated users—typically people who have graduate degrees in computer science or mathematics. The impact of mechanized mathematics on mathematics education will only be marginal until systems are made available that are usable by and useful to ordinary mathematics students.

4. **Division between Theorem Proving and Computer Algebra.** An IML requires the capabilities offered by both theorem proving systems and computer algebra systems. These systems are at different ends of the mechanized mathematics spectrum, and the communities that work on developing them communicate with each other very little. In an IML, theorem proving and computer algebra must be integrated.
5. **Mathematics Library.** An effective IML needs a well-endowed mathematics library. A great deal of mathematics would have to be carefully formalized, mechanized, documented, and interrelated. The development of the library would be a huge task, and there is no consensus in the mathematics and automated deduction communities for how the library should be organized and constructed.
6. **System Design.** To be effective, the various subsystems of an IML (e.g., user interface, logic, prover, calculator, and theory development facility) must be highly integrated like the component systems of an automobile. System developers in the field of mechanized mathematics have relatively limited experience in the kind of system design needed for an IML.
7. **Underlying Logic.** The great majority of contemporary theorem proving systems are based either on simple logics which are too inexpressive or on type theories which are too esoteric for most typical mathematics practitioners. To be successful, an IML needs an underlying logic which is highly expressive, familiar to users, and well understood by developers. We have proposed a logic called STMM [8, 9] that is intended to serve as a foundation for mechanized mathematics. STMM is a version of von-Neumann-Bernays-Gödel (NBG) set theory [13, 18] with convenient machinery for reasoning with undefinedness and partial functions.

5 Proposal

It will take a tremendous effort to make the idea of an IML into a reality. It is highly unlikely that one organization or research group could overcome the obstacles given above sufficiently to develop an IML by itself. On the other hand, the potential benefits of an IML for mathematics education are immeasurable.

We propose that the automated deduction community should make the development of an IML for mathematics education its foremost goal and should seek support for the pursuit of this goal from both the mathematics research and mathematics education communities. The fulfillment of this goal would revolutionize mathematics education, making it possible for many more people to learn, do, and appreciate mathematics. It would also convincingly demonstrate that automated deduction is a valuable and important technology for our society.

References

1. B. Barros et al. The Coq proof assistant reference manual, version 6.1. Available at `ftp://ftp.inria.fr/INRIA/coq/V6.1/doc/Reference-Manual.dvi.gz`, 1997.

2. J. Barwise and J. Seligman. *Information Flow: The Logic of Distributed Systems*, volume 44 of *Tracts in Computer Science*. Cambridge University Press, 1997.
3. R. Boyer and J Moore. *A Computational Logic Handbook*. Academic Press, 1988.
4. B. W. Char, K. O. Geddes, G. H. Gonnet, B. L. Leong, M. B. Monagan, and S. M. Watt. *Maple V Language Reference Manual*. Springer-Verlag, 1991.
5. R. L. Constable, S. F. Allen, H. M. Bromley, W. R. Cleaveland, J. F. Cremer, R. W. Harper, D. J. Howe, T. B. Knoblock, N. P. Mendler, P. Panangaden, J. T. Sasaki, and S. F. Smith. *Implementing Mathematics with the Nuprl Proof Development System*. Prentice-Hall, Englewood Cliffs, New Jersey, 1986.
6. D. Craigen, S. Kromodimoeljo, I. Meisels, B. Pase, and M. Saaltink. EVES: An overview. Technical Report CP-91-5402-43, ORA Corporation, 1991.
7. W. M. Farmer. Theory interpretation in simple type theory. In J. Heering et al., editor, *Higher-Order Algebra, Logic, and Term Rewriting*, volume 816 of *Lecture Notes in Computer Science*, pages 96–123. Springer-Verlag, 1994.
8. W. M. Farmer. A proposal for the development of an interactive mathematics laboratory for mathematics education. In E. Melis, editor, *CADE-17 Workshop on Deduction Systems for Mathematics Education*, pages 20–25, 2000.
9. W. M. Farmer and J. D. Guttman. A set theory with support for partial functions. *Studia Logica*, 66:59–78, 2000.
10. W. M. Farmer, J. D. Guttman, and F. J. Thayer Fábrega. IMPS: An updated system description. In M. McRobbie and J. Slaney, editors, *Automated Deduction—CADE-13*, volume 1104 of *Lecture Notes in Computer Science*, pages 298–302. Springer-Verlag, 1996.
11. W. M. Farmer, J. D. Guttman, and F. J. Thayer. IMPS: An Interactive Mathematical Proof System. *Journal of Automated Reasoning*, 11:213–248, 1993.
12. W. M. Farmer, J. D. Guttman, and F. J. Thayer. Contexts in mathematical reasoning and computation. *Journal of Symbolic Computation*, 19:201–216, 1995.
13. K. Gödel. *The Consistency of the Axiom of Choice and the Generalized Continuum Hypothesis with the Axioms of Set Theory*, volume 3 of *Annals of Mathematical Studies*. Princeton University Press, 1940.
14. M. J. C. Gordon and T. F. Melham. *Introduction to HOL: A Theorem Proving Environment for Higher Order Logic*. Cambridge University Press, 1993.
15. R. D. Jenks and R. S. Sutor. *Axiom : The Scientific Computation System*. Springer-Verlag, 1992.
16. Macsyma. *Macsyma Mathematics and System Reference Manual*. Macsyma Inc., 1996.
17. W. McCune. OTTER 2.0. In M. E. Stickel, editor, *10th International Conference on Automated Deduction*, volume 449 of *Lecture Notes in Computer Science*, pages 663–664. Springer-Verlag, 1990.
18. E. Mendelson. *Introduction to Mathematical Logic*. Van Nostrand, 1964.
19. R. P. Nederpelt, J. H. Geuvers, and R. C. De Vrijer, editors. *Selected Papers on Automath*, volume 133 of *Studies in Logic and The Foundations of Mathematics*. North Holland, 1994.
20. T. Nipkow and L. C. Paulson. Isabelle-91. In D. Kapur, editor, *Automated Deduction—CADE-11*, volume 607 of *Lecture Notes in Computer Science*, pages 673–676. Springer-Verlag, 1992.
21. S. Owre, S. Rajan, J. M. Rushby, N. Shankar, and M. Srivas. PVS: Combining specification, proof checking, and model checking. In R. Alur and T. A. Henzinger, editors, *Computer Aided Verification: 8th International Conference, CAV '96*, volume 1102 of *Lecture Notes in Computer Science*, pages 411–414. Springer-Verlag, 1996.

22. P. Rudnicki. An overview of the MIZAR project. Technical report, Department of Computing Science, University of Alberta, 1992.
23. J. R. Shoenfield. *Mathematical Logic*. Addison-Wesley, 1967.
24. S. Wolfram. *Mathematica: A System for Doing Mathematics by Computer*. Addison-Wesley, 1991.