Reflections on Modeling and Teaching Modeling[1]

Karl A. Smith and Anthony M. Starfield

Solomon Garfunkel's delightful reflection on the 40th anniversary of COMAP [2022] reminded us that our book, *How to Model It: Problem Solving for the Computer Age* [Starfield, Smith, and Bleloch 1990], has passed its 30th anniversary and is still going strong. We appreciate the opportunity to reflect on the ideas that prompted the book, our modeling philosophy, and their implications and applications. These ideas have, of course, evolved since the book was written, becoming more organized and explicit. We hope that they will help reinvigorate the movement to emphasize the development of modeling *thinking*.

We focus first on the background and philosophy behind the book and our thoughts on the principal aspects and core features of modeling.

We then discuss our pedagogical approach to developing modeling thinking skills and their implementation in mathematical and computational thinking courses.

Finally, we offer some insights and lessons learned.

Background and Philosophy

Our book and the modeling courses that we have taught for many years are based on a pragmatic modeling philosophy. The overarching idea is the dichotomy between the "real world "and the "model world." The "real world" is out there all around (and inside) us; we understand it imperfectly. We build "model worlds" implicitly in our brains so that we can operate in the real world. An *explicit* model, one that we can share with others, is what we mean when we talk of modeling.

So, a model is an intellectual tool (that operates in a "model world"), just as a spade is a gardening tool. And just as we don't have a universal gardening tool, but a shed full of different tools, we construct a wide range of models, each for a specific purpose.

It follows that a model is a *purposeful* tool. To set out to build a model without a clearly articulated purpose is equivalent to performing an experiment without a hypothesis. In fact, the analogy with experiments is useful because an experiment (even one in the field) also operates in a model world, and we must interpret its results carefully in light of the constraints and assumptions inherent in its original design. A model, like an experiment, can help us to understand the original question better, or make a personal or communal decision, or suggest new hypotheses.

People who create or use models tend to focus on the inner workings of the model or the results obtained from it. These are, of course, important; but even more important are

---

[1] Prepared for The UMAP Journal, The Journal of Undergraduate Mathematics and Its Applications

- the *purpose* of the model,
- the *design* of the model world that lies behind the actual working model,
- the *assumptions* inherent in that design, and
- the *limitations* of the data fed into the working model, and consequent limitations of the model.

Always remember that any conclusion that you draw from a model is an IF–THEN statement: IF your stated purpose is relevant, and if you make these assumptions and use these data, THEN….

This constant awareness of the limitations, weaknesses, and assumptions of a model is what makes modeling powerful and distinguishes the explicit models that we design from the implicit models in our brains.

Putting this philosophy into practice leads to a modeling paradigm, built on **rapid prototyping**, which goes something like this:

- Define your **purpose**.
- Design the **simplest** model world that you can imagine for that purpose; this is your first prototype. A classic example of this is John Harte's Consider a Spherical Cow [Harte, 1985].
- List your **assumptions**.
- Develop the inner workings of your **model**. (This is where one might eventually use a mathematical solution or coding software, but one does not want to be locked into a solution method during rapid prototyping, so Excel might be the best way to get started.)
- Use the **best data** that you have for model parameters and your best guesses where data are missing.
- **Run your model** and try to **interpret** your results back from the simple model world to the complex real world.
- Do an **assumption analysis** (our terminology) to explore how sensitive your conclusions are to your assumptions—in other words, tweak your model imaginatively to estimate what practical difference your assumptions might make.
- Do a thoughtful **sensitivity analysis** to estimate the practical effect of the uncertainties in your data.
- Review whether the **insight** from the prototype suggests that your original purpose may have been misconceived.

- **Decide** on the basis of your purpose review and assumption and sensitivity analyses whether you need a further prototype and why. Go back, if you do, to the beginning.
- **Repeat** as necessary, keeping each new prototype as simple as you can.

Pedagogical Approach

When we were formulating and teaching a modeling course for first year engineering students in the late 1970s and early 1980s and changing the approach in courses that extensively involved modeling, such as systems analysis and numerical methods, we were also deeply involved in developing and implementing engaged pedagogy, especially cooperative learning [Smith et al. 1981a; 1981b]. We think of learning as requiring **deliberate distributed practice**, that is, it must be intentional and especially considerate of helping learners to manage their cognitive load; it must be distributed over time and approach (i.e., written, oral, visual, kinesthetic, etc.), and it requires practice with feedback [Streveler and Smith 2020]. Remarkably, more than 30 years ago we were embracing these ideas; however, we hadn't explicitly articulated some of them.

A critical first step is identifying *enduring outcomes*, that is the knowledge, skills, and attitudes or habits of mind that you want the learners to master and retain long after the course has ended. On the surface, this seems straightforward; however, to differentiate the enduring outcomes from important-to-know outcomes and nice-to-know outcomes is very difficult. Here are a couple of the salient overall goals and expectations from the syllabus of our early courses:

- Learn about formulating, modeling, and analyzing engineering problems
  - Master the concepts, principles, and heuristics
  - Develop skills for formulating and solving problems
- Improve skills for using tools (computers) for modeling and problem solving
  - Develop proficiency using the application software tools

If we were designing and teaching a course today, we would more carefully articulate enduring outcomes and frame those outcomes using either the revised Bloom Taxonomy [Anderson and Krathwohl 2001] or the Fink Taxonomy of Significant Learning Outcomes [Fink 2013]. One nice feature of the Fink Taxonomy is that in addition to including cognitive outcomes, it also includes affective and meta-cognitive outcomes.

One of the key pedagogical features of all our courses is the emphasis on *problem-based* learning and *project-based* learning; that is: *The problem always comes first.* We use the problem to motivate and direct learning [Smith and Starfield 1993]. We periodically reminded learners of key heuristics, such as: What is the purpose? How good an answer is needed?

We originally experimented with a variety of approaches, including using a case study approach in a systems analysis course; however, after trying them, we shifted to using moderately complex open-ended problems [Smith, Wassyng and Starfield 1983; Starfield and Smith 1988]. The 1988 paper includes a draft chapter from *How to Model It,* and shows how we attempted to implement, in a book, the practice of open-ended problem solving, reflection, and interactive learning. One can even create imaginary model worlds that are carefully designed to introduce students to modeling concepts [Starfield and Salter 2010].

Specifically, we used the pedagogical approach of *cooperative* problem-/project-based learning (CPBL/CPrBL), which are two kinds of challenge-based learning [Bransford et al. 2002]. The format for both is similar; however, the goals are different. In problem-based learning, the goal is to formulate and solve the problem, whereas in project-based learning, the goal is to complete the project. This kind of interactive learning requires a great deal of implementation planning to execute and evaluate effectively, on which we now elaborate.

- Task: Formulate and solve the problem(s) (or complete the project).
- Individual: Develop ideas, approaches, alternatives, initial models and estimates. Note potential strategies.
- Cooperative: The goal is a single set of answers (or one project report). Strive for agreement.
- Expected Criteria for Success: Make sure that everyone participates and can explain the model and the strategies used (or the project details) to develop it, as well as the reasons for the choice of the design of the model.
- Evaluation: Internal—best answer/design within the available resource or constraints. External—feedback based on rubrics.
- Individual Accountability: Individual assessments, e.g., exams, quizzes, written products. One member from the group may be chosen to explain how the group solved the problem (or to give the rationale for design decisions).
- Expected Behaviors: Active participation, checking, encouraging, and elaborating by all group members.
- Intergroup Cooperation: When it might be helpful, check procedures, answers, and strategies with another group.

For further information on cooperative problem- or project-based learning, see Mohd Yusof et al. [2011], Smith [2000], and Smith and Felder [2023].

Although it requires time, care, and attention to implement cooperative problem-based learning, the approach is very effective in helping learners master knowledge, skills, and develop attitudes that will serve them very well in their studies and careers.

Insights and Lessons Learned

We impart some insights and lessons we have learned from decades of designing and teaching modeling courses:

- **Why modeling?** Models are to be found everywhere in the curriculum (from the Bohr model of the atom to predator-prey models in ecology), in research and in daily life (for example, weather prediction or the spread of pandemics). Our primary purpose for helping students learn how to model is for *understanding*. When students construct a purposeful representation, i.e., a model, they develop a better understanding of the phenomenon that they are modeling. They develop skills and confidence for figuring things out (another key goal). They learn to critique and compare models, to present and explain models, and evaluate the conclusions drawn from other people's models. They should be able to engage more effectively with models they subsequently meet as students, professionals or even as responsible citizens.

- **When should it be taught?** It follows from the previous paragraph that starting early in a student's post-secondary career is important. Math professors often ask how modeling can be taught without a calculus requirement; the answer is very easily. Moreover, modeling can motivate students who are required to take calculus or other math classes.

- **How to help students learn to model?** Teaching and learning constructs that have worked for us include:

  - **Begin with a problem**, not with a theory or algorithm. Depending on the experience of the students—especially for first-year students—starting with a somewhat (but not too) complex and open-ended problem is best. Let the problem drive the process of framing and formulating the model(s), as well as the decisions, including the critical one of how good an answer is needed.

  - Use a **reflective and interactive process** to help students learn important heuristics for modeling, such as: keep it simple, use salami tactics (i.e., divide problems into smaller chunks), etc.

  - Have students work **cooperatively in teams**.

In conclusion

One can tailor modeling in different ways, in different contexts, with varying goals. One goal might be to motivate mathematics, which would have an emphasis on formal rather than computational solutions. In engineering the focus might be on design; in business, political science, and resource management (for example) on decision-making. Just as a model is meaningless sans purpose, so it is essential to be very clear on the purpose (the overarching student learning and development outcomes) of a modeling course.

We emphasize the importance of teaching "pure" modeling as the primary focus [Starfield and Salter 2010; Mason et al. 2014], which leads to the conclusion that *modeling can be* *taught anywhere to anybody*. *A pure modeling course is more fundamental than calculus*; and as with calculus, disciplines can build on the foundation.

We thank Kayla Blyman for suggesting that we write this reflection and very much appreciate editorial assistance from Paul Campbell.

References

Anderson, Lorin W., and David R. Krathwohl (eds.). 2001. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. New York: Longman.

Fink, L. Dee. 2013. *Creating Significant Learning Experiences: An Integrated Approach to Designing College Courses, Revised and Updated*. San Francisco: Jossey-Bass. https://books.google.com/books?hl=en&lr=&id=cehvAAAAQBAJ&oi=fnd&pg=PT9&dq=fink+2013+taxonomy&ots=GDmHvQ8rFJ&sig=9IRCzoE3zRa5xT8HeslHYyS2leQ.

Garfunkel, Solomon A. 2022. COMAP at 40+ and still going strong. *The UMAP Journal* 42 (2): 93–102.

Harte, J. 1985. *Consider a spherical cow: A course in environmental problem solving*. Los Altos, CA: Kaufmann.

Mason David P., Michael Sears, and Anthony M. Starfield. 2014. Ubiquitous modelling: In honour of Tony Starfield's 70th anniversary. *South African Journal of Science. 110(5/6),* Art. #a0067, 5 pages. http://www.scielo.org.za/scielo.php?script=sci_arttext&pid=S0038-23532014000300008&lng=en&nrm=iso

Mohd-Yusof, K., Helmi, S., Jamaludin, M. Z., & Harun, N. F. 2011. Cooperative Problem–Based Learning (CPBL): A practical PBL model for a typical course. *International Journal of Emerging Technologies in Learning, 6*(3), 12–20. Retrieved from https://www.learntechlib.org/p/45159/.

Smith, Karl A. 2000. Going deeper: Formal small-group learning in large classes. Chapter 3 in *Strategies for Energizing Large Classes: From Small Groups to Learning Communities: New Directions for Teaching and Learning*, no. 81, edited by Jean MacGregor, James Cooper, Karl Smith, and Pamela Robinson, 25–46. San Francisco: Jossey-Bass. https://karlsmithmn.org/wp-content/uploads/2017/08/NDTL81Ch3GoingDeeper.pdf.

Smith, Karl A. and Richard M. Felder. 2023. Cooperative Learning in Engineering Education: The Story of an Ongoing Uphill Climb. In Robyn Gillies, Barbara Millis, and Neil Davidson, eds. Contemporary Global Perspectives on Cooperative Learning. New York: Routledge.

Smith, Karl A., David W. Johnson, and Roger T. Johnson. 1981a. The use of cooperative learning groups in engineering education. *Proceedings of the Annual ASEE/IEEE Frontiers in Education Conference*, Rapid City, SD, October 1981, 19–21.

Smith, Karl A. , David W. Johnson, and Roger T. Johnson. 1981b. Structuring learning to meet the goals of engineering education. *Journal of Engineering Education* 72 (3): 221–226.

Smith, Karl A., and Anthony M. Starfield. 1993. Building models to solve problems. In *Teaching Critical Thinking: Reports from across the Curriculum*, edited by John H. Clarke and Arthur W. Biddle, 254–263. Englewood Cliffs, NJ: Prentice-Hall. https://karlsmithmn.org/wp-content/uploads/2017/08/Smith-Startfield-Modeling Engineering Problems-Biddle_Clarke-1993.pdf.

Smith, Karl A., Alan Wassyng, and Anthony M. Starfield. 1983. Development of a systematic problem solving course: An alternative to the use of case studies. *Proceedings of the Annual ASEE/IEEE Frontiers in Education Conference*, 42-46. https://karlsmithmn.org/wp-content/uploads/2017/08/Smith-Wassyng-Starfield-Development-of_a_systematic_problem_solving_course-FIE-1983.pdf

Starfield, Anthony M, and Karl A. Smith. 1988. Open-ended problem solving via punctuated dialogue. *Proceedings of the Annual ASEE/IEEE Frontiers in Education Conference*, 183–192. https://karlsmithmn.org/wp-content/uploads/2022/12/Starfield-Smith-Open-ended_problem_solving_via_punctuated_dialogue-FIE-1988.pdf

Starfield, Anthony M, Karl A. Smith, and Andrew L. Bleloch. 1990. *How to Model It: Problem Solving for the Computer Age*. New York: McGraw-Hill. 1994. Reprint. Edina, MN: Burgess International Group. 2020. Reprint. Buford, GA: LAD Custom Publishing. https://ladbookstore.com/products/how-to-model-it-problem-solving-for-the-computer-age-print.

Starfield, Anthony M, and Richard M. Salter 2010. Thoughts on a general undergraduate modelling course and software to support it. *Transactions of the Royal Society of South Africa*, 65 (2): 116-121. http://dx.doi.org/10.1080/0035919X.2010.513172

Streveler, Ruth A., and Karl A. Smith. 2020. Opinion: Course design in the time of coronavirus: Put on your designer's CAP. *Advances in Engineering Education* 8 (Fall) (4). https://advances.asee.org/wp-content/uploads/Covid 19 Issue/Text/AEE-COVID-19-Ruth.pdf.

Karl A. Smith: Karl Smith is Emeritus Cooperative Learning Professor of Engineering Education, School of Engineering Education, at Purdue University. He is also Morse-Alumni Distinguished University Teaching Professor and Emeritus Professor of Civil, Environmental, and Geo- Engineering at the University of Minnesota. He joined the University of Minnesota in 1972 and started his academic career as a materials processing engineering researcher. In 1991 he changed careers to focus on engineering education research and in 2006 he accepted a part time position as Cooperative Learning Professor, School of Engineering Education, Purdue University to help start the engineering education PhD program in the College of Engineering. His research and development interests include building research and innovation capabilities in engineering education; faculty and graduate student professional development; the role of cooperation in learning and design; problem formulation and modeling; and project and knowledge management. Karl adapted the cooperative learning model to engineering education and has over 40 years of experience working with faculty to redesign their courses and programs to improve student learning. His work on cooperative learning has helped thousands of faculty members build knowledge, skills, and confidence for involving their students in interactive

and cooperative learning both during class time and outside of class. The effects of the work are significant in terms of creating a sense of belonging and membership in a community, as well as much more engaged and deep learning. He recently has focused on high-performance teamwork through his workshops and book *Teamwork and Project Management* (2014). His bachelor's and master's degrees are in metallurgical engineering from Michigan Technological University and his Ph.D. is in educational psychology from the University of Minnesota.

Anthony M. Starfield: Tony Starfield started out with an undergraduate degree in Applied Mathematics and an intention to do graduate work in numerical analysis. Serendipity diverted him from theory to practical applications and a PhD thesis in engineering, all at the University of the Witwatersrand in South Africa. Serendipity (again) took him to the University of Minnesota and academic appointments in Geo-mechanics, interrupted by nearly a decade (in the 1970s) as the Professor of Applied Mathematics and head of a department that included applied math, computer science and statistics at his alma mater. When he retired from academic life in 2005, he was a distinguished professor in the department of Ecology, Evolution and Behavior at the University of Minnesota. He has authored or co-authored over 100 papers and 4 books. Underlying these seemingly disparate activities is a more coherent story: It was exciting for an aspiring applied mathematician to be exposed to "real world" problems in the early days of computing (the first half of the 1960's); the challenge was not only to develop numerical approaches that would have been impractical previously, but also to think about how to pose problems and present results in ways that made a difference to practicing engineers. Applied mathematics was morphing into modeling and problem-solving, and this opened up new areas of application which in turn stimulated new ideas about modeling and teaching modeling. Modeling approaches borrowed from engineers and physical scientists do not always transplant well. His book *Building Models for Conservation and Wildlife Management* (1986) with Andrew Bleloch develops a rather personal approach to ecological modeling. Teaching modeling is different from teaching pre-computer applied mathematics; towards the end of his academic career and post-retirement, Tony was able to think about and develop workshops and one-week classes in modeling and structured decision making for professional scientists and managers in federal agencies such as the Forest Service and Fish & Wildlife Service. Working from the hypothesis that modeling is a creative, logical subject rather than a mathematical one, and that it is learned (by doing) rather than just taught (through lectures) he has also enjoyed helping undergraduate students, with majors ranging from computer science to politics, to develop modeling skills.