# NSF PROPOSAL: EVALUATION AND DISSEMINATION OF TASK-ORIENTED MATH COURSEWARE

FRANK QUINN

## Introduction

This proposal is for a one–year effort to assess and make available materials developed over the last decade. These materials are mainly at the college level. The main objective (and urgency) is to provide a solid base for development of K–12 materials.

An overview of the task–oriented approach is given in the first section. In brief, it depends on high–quality computer-based tests with auxiliary materials, designed to function as learning environments. This is the basic model for most courses at the Math Emporium at Virginia Tech. This facility is in its twelfth year of operation and serves more than 5,000 students each semester.

There are two components of the proposal. The first, in §2, is for rigorous evaluation of a task-oriented course. The second, in §3, is for development work on components of an appropriate next-generation test system. A key feature is nearly complete separation of content and presentation from distribution and administration.

If this proposal is funded then a longer-term and larger-scale proposal concerned with K–12 development and starting in 2010 is planned. Some components of this development, e.g. the American Math Society Working Group on Technical Careers, have already been started.

## 1. Task-oriented courseware, an overview

Task–oriented courseware approaches math education through computer–based problem sets. The approach was developed by observing how students try to use computer–based materials and then optimizing the materials to be used that way. It was not derived from an abstract conceptual construct, and was not modeled on classroom practices.

Students see task problem sets as tests, or practice versions of a test. Students having this view does not cause difficulties, and indeed provides good motivation. It is essential, however, that educators not see these as being essentially tests. Learning is the primary goal. Features needed for learning are quite different from those sufficient for assessment, and approaching this from an assessment background or point of view will not give good outcomes. We do not use the term "formative assessment", for example, because in standard use it is still understood as an assessment.

*Date*: March 18 2009, revised July 2009.

The course described further in the next section provides a bare–bones example. This is a one–semester course with six tests (more precisely, task problem sets) and a final exam. The tests can be taken up to five times for credit, with the best score counted, and unlimited practice versions are available. In most sections the tests provide the only assessments:

- Homework is not assigned. Most students do far more practice work on practice tests than they would on traditional homework. Further they try to do test-quality work rather than homework-quality, and don't have to be bribed with credit to do it.
- Quizzes are generally not given. Some instructors have tried them and they don't seem to effect outcomes.
- The "tests" are designed as learning environments. Problem-specific diagnostic aids (usually an intermediate step) are provided to help students locate errors, and web links lead to brief and targeted reference materials.

This approach is almost opposite to current promotion of computer systems for homework and quizzes. The point is that a genuinely effective learning–oriented approach to high-stakes testing renders homework and low-stakes testing unnecessary.

Use of this approach for high-stakes K-12 tests could transform US math education, and this is the eventual project goal. This is discussed in the essay Beneficial High–Stakes Math Tests: An Example. Use in courses is discussed at some length in Task–oriented Math Education. However the first step is to demonstrate that the approach really does work.

## 2. Proposal Part I: a course evaluation

The first request is for support for downstream analysis of a task–oriented course. Most of the data has already been collected and will be analyzed over the summer whether the project is supported or not. Support would enable considerably more detailed and sophisticated analysis, and careful preparation for publication.

2.1. **The Course.** Math 1206 is the second semester of first-year calculus for science and engineering. It is a relatively thorough treatment of techniques of integration, but the main learning goals are analytical and problem-solving skills rather than integration per se. The PI developed learning-oriented computer tests for this course during the 2003-04 school year. The materials stabilized and were used in multiple sections beginning in Fall 2004.

- The study period is the five-year period Fall 2004 - Spring 2009.
- During this period nearly 9,000 students enrolled in the course. Approximately half of these were in computer-tested sections, half in traditional classes.
- Computer and traditional sections used essentially the same syllabus, the same text, and all took the same common final exams.

Final exam data was tracked on an ongoing basis and showed no essential difference between computer and traditional sections, see Integral calculus outcomes, Spring 2006. However this is less impressive than it sounds. The exams are supposed to be consistent and appropriate to the course but there is no outcomes analysis or feedback. Actual scores are often low, vary considerably year-to-year, and correlate only weakly with tests or final grades in either computer or traditional sections.

The careful conclusions from exam data are that it raises no red flags and suggests computer students are learning integration about as well as those in traditional sections. This data does not show that they are meeting the deeper learning goals. This is the objective of the downstream evaluation.

This course is considered a gateway and is a prerequisite for study in math, engineering and most sciences. (There is a less-demanding course for biology.) Va Tech is the largest state university in Virginia and has the flagship science and engineering programs so this is taken very seriously.

- No content compromises were made in the computer sections, as is so often the case.
- There should be significant downstream consequences to success or failure of the course design.
- Almost all of the students in the study period have taken later courses and nearly half have graduated.

Downstream performance should therefore provide a definitive test of the methodology and materials.

2.2. **Status and possible outcomes.** The study has been authorized by our Institutional Review Board[1]. The Registrar, after some hand-wringing, has agreed to provide data on *all* courses taken by these students. Most of the data up to this semester has been provided for preliminary analysis. There are nearly three hundred thousand records in the initial data set. Final data, including this semester, will be provided early in the summer.

It is already clear that extracting a good picture will take careful, thoughtful work.

- The data distributions are not normal. In particular standard correlation analysis is too sensitive to outliers, and culling outliers is too likely to give outcomes that reflect the culling procedure.
- Course effects seem to be second-order, dominated by student characteristics (e.g. strong and ambitious students do well no matter what).
- Many important characteristics such as work habits and personality types are not measured and seem to have unexpected or conflicting consequences.

An overview of pitfalls and common errors in educational data analysis is given in Evaluation of methods in math education.

Support will enable us to do cluster analysis of outcomes in major science and engineering courses. This should help identify student sub-populations with consistent characteristics, and will be of independent interest.

The most likely outcome is that the software is not yet the equal of traditional teachers. Reasons to expect this include:

- There are some truly remarkable teachers in the traditional sections while the software, though excellent in basic design, is incomplete and far from state-of-the-art.
- The syllabus and course pacing was inherited from the traditional sections and the data already shows they are not well-adapted to the computer sections (the first two tests are too easy in comparison to the last four).

[1] IRB # 08-714, approval date 12/4/2008

- Computer-tested sections are two or three times as large as traditional sections because there is no grading burden. We would expect weaker outcomes on the basis of class size alone.

However the real goal is to assess how far the software has to go and guide further development, not decide if it should be discontinued. Budget pressures will probably force use of computer testing in *all* sections in the very near future.

## 3. Proposal Part II: next-generation test system components

The educational community needs a new approach to computer-based testing. In particular, content needs to be thoroughly separated from the mechanics of test administration.

### 3.1. **Design criteria.**

- Content development should be a community activity and content should be portable, not restricted to particular systems.
- Question and answer types should be driven by educational concerns, not restricted by programming decisions in system design.
- In particular, problems designed to promote learning require more complex and imaginative answer formats than do pure assessment tests.
- More than 90% of tests served in a task-oriented system are used for practice. It is undesirable and unnecessary to have these materials tied to a server.

Another important goal is a shift from databases to software-generated materials. Not long after the Emporium opened in 1997 the PI developed ways to generate problems directly with software rather than drawing them from a database. See Software-generated test questions for an early (2001) description of the approach. The current Math Emporium test system now uses this approach and has served over a million tests. Software generation can provide significantly higher quality in a variety of ways: roughly speaking it is possible to capture educational wisdom and content sophistication in problem-generating code. The calculus course described above depends heavily on this.

Current test systems are strongly database-oriented. It seems to be infeasible to convert them to use software-generated materials, let alone to flexible standard formats.

3.2. **PDF tests.** We propose developing a system in which a *test* (or learning task) is an Adobe PDF document:

- Embedded Javascript can be used to make it self-scoring;
- Upon scoring it should provide answers, diagnostic aids, web links, and other learning aids.
- Only for-credit tests should be tied to a server, for authentication, recording grades, etc.

This approach allows for-credit administration systems to be designed or adapted for specific needs (state K-12 tests, online college courses, etc.) but all draw on a general pool of content generators.

In fact self-scoring PDF tests can be constructed with off-the-shelf software:

- D.P.Story's AcroTeX eDucation bundle provides a LaTeX package for embedding Javascript self-scoring functionality.

- LaTeX code with appropriate markup is processed directly to PDF by pdf-TeX or through PostScript and the Acrobat Distiller.

Story's materials are designed for hand-crafting individual problems and tests rather than for large-scale use. Variations have been developed by R. Moore and F. Griffin (Macquarie U.) and others. The Griffin-Moore system uses software generated problems with generators written in Perl.

3.3. **A TeX working group.** Story's materials provide a strong foundation but need to be updated, better documented, and adapted to large-scale use. The use of TeX as a base suggests a strategy: LaTeX and other extensions are open software projects with strong volunteer participation. A Technical Working Group of the TeX Users Group devoted to TeX-based tests may attract volunteer contributions. It would also permit identification of experts and provide a context for supported work if support were to be available.

The proposal is to organize a Technical Working Group and launch it with a critical mass of documentation of current materials, samples, and a clear description of what is needed. Officers of TUG are receptive to the idea and have invited an article describing the effort for the next issue of the TeX User's Group journal.

3.4. **A web browser.** Another ingredient is a web-based browser that can be used to explore output of problem generators with different parameter settings. We already have a good single-user browser for developing test specifications. It will be straightforward to modify this for web use when the output is a PDF file.