Functioning in the Wireless Classroom

Shelley Goldman, Roy D. Pea, Heidy Maldonado, Lee Martin, Toby White

Functioning in the Wireless Classroom

Shelley Goldman, Roy Pea, Heidy Maldonado, Lee Martin, Toby White
and the WILD Team @ Stanford University [1]

Abstract

Code It! fosters mathematics learning environments where pre-algebra students use handheld technologies to confidently and enjoyably explore and learn about functions. The resources we developed—server-based and handheld software and paper-based student and teacher texts—were packaged as a 20-session unit on code making and breaking and designed to boost students' understanding of mathematical functions and their facility with the multiple representations of tables, graphs and symbols. We field tested the wireless system with two teachers and 120 students during summer school, and conducted studies on the features and function of the technology as a learning and teaching resource. We report on project development and research, focusing on lessons learned about the strengths and difficulties of wireless, handheld technology in the mathematic classroom.

1. Need

This is a particularly important time to develop and research technologies for math learning. Across the US, there is a push to raise lukewarm achievement results on standardized tests and international comparisons. Teachers are being asked to teach more content while raising the level of achievement, but are left unprepared. The *Code It!* materials—technology and real world based—were designed to address this challenge, capitalizing on developing technology for stimulating high performance while providing a satisfying, successful learning experience on the topic of functions.

Technology should be providing tools to support the activities of mathematics classrooms. Gathering data, looking for data patterns, developing algorithms, thinking spatially, working with multiple representations, and complex calculations are all greatly facilitated by computers, yet many problems have accompanied classroom computer use: prohibitive cost, use models that do not mesh with classroom structure, and a learning curve for teachers that has been too high for too little gain. Even when researchers developed compelling exemplars [1,2], changes in organizational, social, and pedagogical practices with computers produced high hurdles [3,4]. The combination of increased access, increasing teacher knowledge, decreasing technology costs, and new portable devices is creating a synergy and new possibilities for mitigating problems.

Early studies of handheld technology use are modest yet suggestive of how they might best be applied [5, 6, 7, 8, 9, 10] These are promising trends, and to tap the potential in *Code It!*, we developed technology for enhancing students' math learning by providing representations that can be easily manipulated, and by creating a context for learning formalisms [11,12]. Our project addresses three main research areas:

1.1. Enhanced mathematics learning. Do the materials and wireless tools enhance student learning of crucial algebra concepts and skills? What progress do students make in reasoning with representations of patterns and functions? In communicating mathematically? Are there any increases in productivity with this approach? Is engagement with math high? Do the materials boost student achievement?

1.2. Productive tool use. How well do wireless handhelds or desktop computers work as a tools for math learning? What are the resources and obstacles to use of the handhelds, network infrastructure, and collaborative activity structures for learning mathematics?

1.3. Enhanced teaching. Through partnerships with teachers we learn about conditions for success and needs for further development. Will teachers benefit from unit teaching tools and access to server-based group and individual portfolios and data? What are the critical issues in teacher learning and appropriation of this technology?

We anticipate that the results in each of these research areas will influence the design of a generation of math tools by indicating ways that server-based communication, data tracking and storage capabilities can enhance the teacher's role.

[1] WILD Team@Stanford members contributed to the *Code It!* work reported herein. Thanks to Sarah Walter, Gloria Miller, Mike Mills, John Murray, Nicolai Scheele and Wolfgang Effelsberg. Special thanks to our partnering teachers and their students, the Stanford Teacher Education Program and the Stanford student teachers, and the Santa Clara School District. This project was supported by Stanford Center for Innovation in Learning (SCIL) and the Wallenberg Global Learning Network. We also thankfully acknowledge an HP mobile wireless computing equipment grant to SCIL.

2. Our Response and Approach

The project was structured around three strands of activity: materials development, work with teachers, and field test research in the classroom. Each is discussed in turn.

2.1 Materials development.

We created a technology-integrated 20-session curriculum unit on linear and quadratic functions, with a code making and breaking theme. The unit was developed to bring alive in classroom activity the functionality of the *Code It!* application with pre-algebra content. The unit places algebra learning in a real-world context—making and breaking secret codes [13]. Students can challenge others to break their codes to determine message content. The materials are structured so that students must examine properties of functions in general, learn to distinguish among "families" of functions (e.g., linear vs. quadratic) as well as learn properties of each family (effects of changes in constants to graph, for example). Connections among symbols, tables and graphs are emphasized.

Code It! was inspired by a curriculum unit, *Codes, Inc*, developed by the Middle-school Mathematics through Applications Project (MMAP), which won recognitions as a standards-based curriculum as ComputerWorld Smithsonian Awards finalist, and as a promising middle school mathematics curriculum as judged by two independent US Department of Education expert panels. Evaluation showed teachers found the MMAP codes unit helpful for transitioning to great use of technology and more applications-based curriculum materials [14]. Teachers saw real values for the unit with its multiple representations of functions to aid learners' development of rigorous algebra skills.

2.1.1 The change from desktop to WILD. The advantages of the wireless version of the software over the desktop version are obvious. With hand-helds and web technologies, students can create and exchange codes easily, and records of their work and communicative transactions can be captured on the server. It is this promise of engaging technology that promotes increased interactions with mathematics and embedded assessment activities that we hope to promote. In the desktop software developed by MMAP, students could create substitution codes software, view graphs, frequency tables, and patterns in codes, and receive verification of successful decoding based on the expressions used for coding. Previously, exchanging the codes was based on cumbersome e-mail, now supplanted by wireless communications.

Code It! software is built around an in-room wireless network which allows students to work on codes together in groups and to share codes between groups. Each PDA is connected wirelessly to a teacher's station located at the front of the room. The teacher's station runs administrative software that allows the teacher to control and monitor groups in a variety of ways. The teacher can monitor which students are logged into the server, place students into groups, create and distribute practice problems, and open an observer window to display a group's current state. The observer window is particularly important for teachers because the small display size of the handhelds makes it more difficult to monitor group work in a traditional "over the shoulder" manner. In addition, the server logs each group's activity, allowing for a researcher or teacher to reconstruct the precise ways in which students were using the software and solving the math problems. The server also acts as a repository for text and codes created by the teacher and students, providing an easy way to share codes among groups and to distribute practice problems.

Image 1: Function table view of a code solution.

Plain Text:
I SIT BESIDE THE FIRE AND THINK OF PEOPLE LONG AGO, AND PEOPLE WHO WILL SEE A WORLD THAT I SHALL NEVER KNOW.

Function Table:

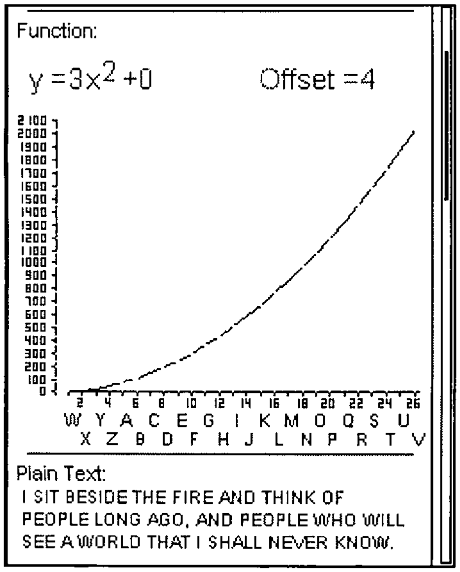
X	Y	X	Y
1	3	14	588
2	12	15	675
3	27	16	768
4	48	17	867
5	75	18	972
6	108	19	1,083
7	147	20	1,200
8	192	21	1,323
9	243	22	1,452
10	300	23	1,587
11	363	24	1,728
12	432	25	1,875
13	507	26	2,028

Each PDA runs *Code It!* software that allows students to create plaintext, to make and break codes, and to upload and download text and codes from the server. When making or breaking codes, the student's screen displays the encoding function (or current guess) and the graphs and tables that represent the function and the coded text.[1]

[1] In the current implementation of the software, these various representations are displayed on a single long screen that must be scrolled up-and-down to bring each representation into view. While this is not ideal, our field-test focus was on ensuring that the software was robust, and the design as it was generally

These representations include a graph of the function, a function table, and word and letter frequency tables. These representations were selected in order to foster particular kinds of learning, rather than to create an ideal coding tool. All of the PDAs within any given group are bound together via the server. A change made by one student propagates to the other students in his or her group, so the PDAs within a group always display the same function and text. Students were encouraged, by the assignment of group roles, to have each group member focus on a different representation. One student can look at the graph, another at the function table, and a third at the letter frequency table, for example. The goal is to foster communication among group members and increase the diversity of information available to the group for the coding or decoding task. When a group has successfully created or broken a code, they submit their solution to the teacher station.

Image 2: Graphical screen view of a code solution.



2.1.2. Instrumentation. Beyond its use for channeling the work of collaborative groups to be in-synch, the server stores all messages being worked on by the different groups. It also provides administrative functions so that the teacher may define groups and assign students to groups. It allows both teacher and students to write and edit plain text messages with a normal keyboard, not just the restricted keyboard of the PDA. The server also logs communications to and from the PDAs to a file that can later be used to recreate the PDA display for any student at any particular time.

2.2. Work with teachers.

The professional development component turned out to be as important as the materials and software in ensuring that students get a well structured, engaging and deep mathematical learning experience through *Code It!*

Teachers "choreograph" the performances of classroom tasks and activities to enable students to learn mathematical ideas and language, and to develop mathematical practices. Teachers are also faced with supplementing their adopted texts to meet students' needs for more in-depth coverage and differing approaches to the most important concepts. They also experiment to increase comprehensive coverage, an emphasis on thinking, problem solving and communication. We know that teachers must do a great deal of work to arrange productive participation structures in their classrooms, and we need to learn about the places that they can find for incorporating wireless technology under daily classroom conditions. This is a complex job, and teachers need time to learn and develop both teaching sensitivities and practices with technology. Their professional development is crucial.

Professional development included three days of training and planning with the teacher partners as well as just-in-time, in-class support. The workshop included:

Time for being introduced to the technology including PDA use and the teacher station.

Early access to review of the curriculum unit, and discussion and revision of each activity.

Practical planning advice and session-by-session planning time for implementing technology in the curriculum. This time included management of technology-related issues such as battery life, the class activity schedule and student roles.

Just-in-time daily consultations on math and technology while implementing the unit.

Our research questions are aimed at uncovering the strategies that teachers use to orchestrate the use of handhelds, identifying the obstacles and affordances they interact with in using *Code-It!* materials and technologies.

2.3. Project questions and methods.

Our research goal was to test the usefulness of the materials and to generate knowledge about the ways in which they were effective. We collected several data streams, including observational data, videotapes of teachers and the four-member student groups used in the classrooms, problem-solving and informational interviews with student dyads, and pre- and post-testing of mathematics knowledge. These methods, of observing and

functioned well. This interface issue will be corrected in future versions.

videotaping classroom field tests of materials and feeding back results into the design, enable the development team to define areas for revision based on the use of the materials under real classroom conditions [15,16,17]. The cycle of development and field test supports our learning about how teachers and students receive materials and how best to iteratively improve the design for achieving the teaching and learning goals.

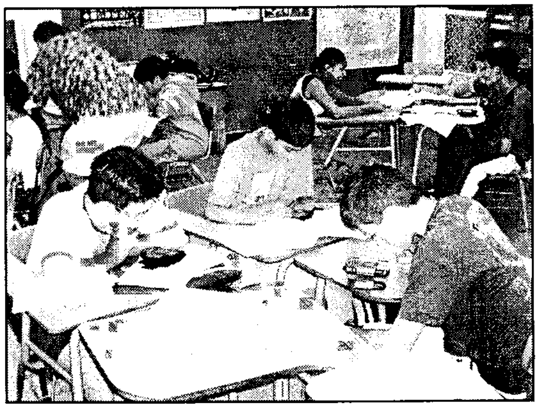
It is part of our method to track topics and issues as they emerge in field tests and teacher workshops, and to then interview, videotape, and interrogate. We look first for obvious patterns of structure such as who is involved in planning and holding activities, how the space is set up and how and when people move around in it, what the flow of activities is from start to finish, what materials and resources are used, how and when teachers and students talk, what they talk about, and how they interpret their experiences after the fact. After searching for some of these basic structures, or because of "noticeable" events, other areas for analysis emerge. These qualitative methods are complimented by the development and use of problem-solving tasks and pre-and post-tests with students and interviews and surveys of teachers.

The research process is characterized by an inter-relationship among field-testing, development, and evaluation. For instance, prospective materials emerge from user tests and focus groups, get written, and get put back into the field test, during which time those materials are evaluated and revised. Finally, materials are put to the test under controlled conditions in a series of summative, clinical studies.

We partnered with the Stanford Teacher Education Program and the Santa Clara, CA Schools to use *Code It!* with all of the district middle school students in their summer school program for our 20-session field test with two teachers over a five-week period. Each teacher had 60 students split between two consecutive 100 minute class sessions: one teacher used handheld iPaq computers (PDAs); the other teacher used Compaq Tablet PCs. Our analysis concentrates on the two PDA classrooms.

Our WILD team developed software and a unit of activities in consultation with the teachers, pre- and post-tested students with mathematics vocabulary items, short answer problems based on unit concepts and representations, and released pre-algebra and algebra items from the California High School Exit Exam. We collected video on 120 hours of classroom time, using two cameras in the PDA classroom each day. This enabled us to "follow" the activities and progress of four groups of students in those classes. We also interviewed twelve pairs of students (N=24) from those focus groups. For students in our focus groups, we have pre- and post-tests, an interview where they were asked to problem solve and break a code, and a video record of their activities and progress through the unit. In addition, the server provided a record of process steps for each group's attempts to make and break codes, and when they successfully completed each task (or not).

Image 3: Student group working on *Code It!*



3. Lessons Learned

Analysis is preliminary, yet we have some encouraging results on how our learning technology design functioned. and the ways in which teaching and learning were affected. Whereas we have outlined the project with three categories of interest, our results reinforce the synergistic nature and interactions of the technology, the teaching and the learning. We do not report on them as if they were mutually exclusive because we observed them in the flow of classroom activities where their affordances and constraints melded into resulting situations and events. So we present here observations where the constellation came together in mutually reinforcing ways or in negative ways. We are able to offer some next steps that we have identified to improve the *Code It!* experience and some of those steps point to specifics of students' math learning, how teacher can make the most optimal use of the materials, and how the design of the *Code It!* technology itself might change.

3.1. Students and math.

Two aspects of student work dominated our attention during and after the field test. The first was meeting our goals of getting students to interact with, discuss, and use the mathematics. The second was the establishment of the social arrangements needed to support working with the tools to meet the math engagement goals.

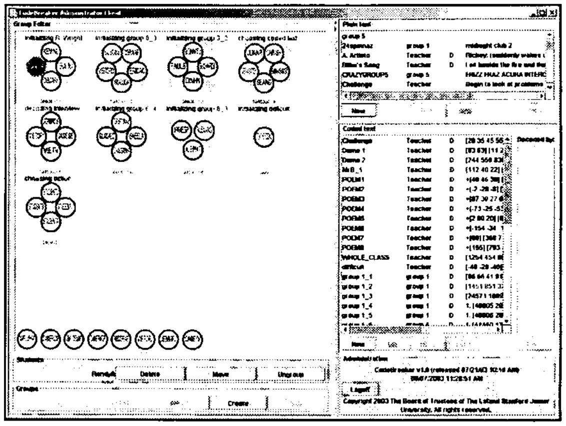
3.1.1. Students' interaction with mathematics. To date, we have analyzed results of the pre- and post-tests in the PDA class with our targeted student groups and the results were very promising (N=45). The mean increase from pre- to post tests was eight percentage points; in four of six focus groups, students made significant gains, in some cases raising their scores by 15-30%. The PDA students showed significant gains on test items relating to evaluating exponents and the graphs of functions. On one graphical item, 44% of students answered correctly on the post-test, as compared to only 13% on the pre-test. What is exciting about these trends is that these results were for students in grades 6-8 who were placed in heterogeneous groups regardless of their previous school math course achievement. Many students in the class had done poorly in sixth or seventh grade math, while some students had completed the first year of algebra. These algebra students were encouraged to help others, and the pre- and post-test results showed that these interactions may have had an impact on achievement.

3.1.2. Focusing on the Curriculum. Turning the attentions of students in the *Code It!* classroom to the mathematics took a great deal of social engineering on the part of the teachers and the development team. The team had to mitigate the effects of the widely heterogeneous groups and the constraints that the *Code It!* tool entered into the equation. We treated these as both challenges and opportunities. During the teacher preparation workshop, we discussed these constraints and decided to introduce rotating social roles inside the groups in order to make sure all students had a chance to work with all aspects, tools and representations during the problem solving process. Once the session began and students started learning how to code and decode using the PDAs, a more anarchistic atmosphere emerged in the classroom. The teacher moved ahead with the activities with a mention of the roles, yet students forged ahead without an obvious group structures.

We saw "stylus wars" break out among students who were vying for control of the group PDAs and communication with the server. Heated discussions arose and the chaos and discontent led us to work with the teaching staff to design social roles that would foster group collaboration, rather than competition. Linking the process roles of (1) recorder, (2) presenter, (3) publisher and (4) equipment manager to the different code representations in the software, and rotating these roles avoided much of the inter-group conflict, but also contributed to more engagement on the part of more students in code analysis and breaking activities. In addition to group roles, the teacher and summer interns instituted and enforced a rule stating that "all students in a group must be able to explain the strategies and routes to solutions in order to receive credit." That meant that if the more skilled and experienced students in the group solved a code before others, the entire group had to review the steps taken, the strategies used, the solution, and be able to demonstrate it on their PDAs. This increased team cooperation, resulted in a practice of questions and explanations across group members. Teams spent time working together to make sure they could each explain their problem solving if asked informally by a teacher or if asked to present to the entire class.

Due to the social engagement rules, we saw more consistency of performances on the part of the students over time. Early on we observed a full range of performances, from students engaging in some extremely rich and iterative collaborative problem-solving processes, to students relying on simple guessing strategies, to students avoiding work altogether. After instituting the social rules for "student roles" and the "explanation rule for credit," new patterns emerged, and we observed multiple instances of the students who took problem-solving approaches talking with those who usually guessed or seemed uninvolved. The social engineering improved group relations and engaged students more deeply with the software and the mathematics under study.

Image 4: Teacher's view (server screen).



3.2. Teachers.

Findings about teachers and their work fall into three categories: (1) teacher training and teacher knowledge; (2) teachers' appropriation of the technology; and, (3) teachers' "workarounds" for the technology's shortcomings.

3.2.1. Teacher training and knowledge. Even though teachers participated in three days of teacher professional development that included explorations of the unit and the technology and individually practiced using the teacher station and the PDAs, the situations that developed when twenty-five to thirty students were using the system in class outstripped their novice knowledge. While the teachers were clear on the basic operations of the PDAs, they were less experienced in working with the interactions that students would practice and the ways the software could be used to pose or solve particular coding problems. As the days of the field test went by all three of these teacher practice areas improved. We did see examples of how continuing work with the teachers on an ad-hoc basis addressed some of the teachers' lack of prior experience. During the first two weeks of the unit, the teachers found themselves grappling with understanding the potential of the technology to help with the students' problem solving processes. In one of several meetings between a math educator on our staff and the teachers, the strategies for working to decode were explored. It was agreed at the meeting that our staff member would pair with one of the teachers so they could demonstrate together a systematic way to use *Code It!* to crack more difficult codes. Once this happened there was an increase in student productivity in code breaking. We realize in analyzing this incident that we were unable to have predicted the teacher's need for scaffolding in order to use *Code It!* most effectively with the students. Next steps will include more classroom problem simulations for the teachers to work with that can be fashioned from this year's field test. This will help teachers to figure out how the tools can best be used for dealing with different kinds of problem solving affiliated with the unit.

3.2.2. Teachers appropriated the technology. The teachers actually did a great deal of work to make the technology useful for meeting the curricular goals. *Code It!* presented them with a set of challenges and complexities to manage and, in some instances, overcome. We saw the teachers appropriate the technology into their teaching practices. One teacher found a way to use the blue and white circles that showed the active status of every student in each group on the teacher station screen as a classroom management tool, keeping track even from across the room of which students were on-line and which groups were most active. This enabled him to troubleshoot for students who were having log-on issues as well as to visit groups that were lagging behind in the decoding problems. The same teacher used the teacher station to develop activities and code breaking problems when he saw the need for more scaffolding between simple and more complex code breaking problems. One student teacher in the classroom also developed problems and entered them into the system through the teacher's station. This gave teachers the opportunity to assess students' work and needs, to create hand-tailored activities and problems, and to organize them so they were available to the students. Both teachers used projection equipment so they and students could discuss with visual aids the various strategies for code breaking as well as demonstrate how the different *Code It!* tools could be used. This demonstrated how the system was robust in use and how teachers made it compatible with their teaching practices.

3.2.3. Teachers worked around technology shortcomings. Some constraints in the technology caused rifts in the smooth flow of classroom activity and the teachers found themselves compensating when needed. The constraints were many and ranged from overload during whole class log-on, to the lengthy scrolling screen with multiple representations of data, the fact that some of the representations of coded data were more useful to students than others (e.g., as the system could only handle short codes they could not benefit much from word frequency analysis data), to the fact that the system could log only group work. The teachers created short codes when they provided supplementary activities, shared strategies for working with all of the representations, reminding students of how to find them, and established and enforced a series of roles and expectations about the sharing of code breaking solutions to mitigate the fact that but one PDA in each group could communicate with the server at a time (and return new results in a graph of the function entered, for example) as *Code It!* tracked only group activity.

Together, these findings about teacher work indicate that, although the technology introduced all kinds of management of instructional activities, the teachers were able to demonstrate a steep learning curve, demonstrate an ability to strategize, plan and respond to student and technological difficulties, and to capitalize on the tools to facilitate math learning.

3.3. Technology design and performance.

Three features of *Code It!* were shown to need reworking during the classroom test: (1) how mathematical representations and tools are organized; (2) the choice to base the system on group activities and interaction over individual acts; and, (3) the design and usability of record keeping functions—potentially so promising with wireless technology. Each is discussed in turn with examples to illustrate why we decided to reconsider the design. Our original goal was to produce a first, yet functional and robust version of *Code It!* in a short development period to have it ready for the summer 2003 classroom test. In essence, we had less than twelve weeks for development. This decision was made because we were particularly interested in partnering with the teacher education program in their summer teaching academy and we were eager to provide a summer school mathematics curriculum that featured wireless computing. To us, this was both a service and an experiment. That time line resulted in several design compromises that raised issues when the system reached the classroom,

3.3.1. The organization of mathematical representations. The ways in which students came to interact with the multiple tools and representations for understanding and finding code keys revealed side effects of our choices. The first representational issue arose from the fact that we designed the application with one, lengthy scrolling screen for different representations instead of using a multiple windows approach. While the scroll function was easy for students to learn and use, it gave prominence to the first representation showing on the screen (the graph), with other representations absent from view without scrolling. We anticipated that giving prominence to the graph would signal a hierarchy to students, and it did. We constructed the curriculum materials and a process for introducing the other tools and representations (a function table, letter frequency and word frequency tables). The group process and daily rotation of assigned roles for students attaching them to each of the representations was a compromise between the constraints of the system and the ways students could best benefit from the representations available. Although the system of each student having a portion of the data displayed for interpretation on his/her PDA made for group collaborative learning, it meant that on any day a student had the opportunity to relate to only one view of the data. We were able to deduce that the multiple representation capabilities of the software is a strength and real resource for mathematical work and learning. A question that now remains for investigation is how structured or unstructured students' access to those representations should be.

This is significant because we also discovered that representations were unequal in their familiarity and usefulness. The graphical tool was extremely powerful because the data window automatically fits to the values of the coded text. This enabled students to do a kind of curve-fitting that usually got them close to a correct code. Many of the students could determine the exponent graphically, and often the lead coefficient as well by adjusting the graph. The function table was just as powerful, though far fewer students learned how to use it well (at least partly because of a programming error that we corrected for halfway through the course). The frequency table was two tools in one, because it had an inverse function table embedded in it. That table was confusing to most students, and even those who had already taken first-year algebra operated with misconceptions when they interpreted it. The word frequency table was used even less due to the fact that our coded text passages were short due to the software slowing down with long passages. Students rarely used it, except to guess which numbers might be representing the single-letter words "I" and "a". These differences in representational placement, form and usefulness had consequences—the students assigned to the graph and the function table had the most interaction with the decoding activity.

It is not clear from our field test that our work on representations was completed to satisfaction. We came away from the field test questioning whether the multiple representations provided an appropriate level of scaffolding for the students' understanding. Some observed groups quickly found simple strategies that led them to crack codes without improving their dominion of functions. For example, some students consistently and repeatedly used the graph of a function as predictors of the range and domain of the encoding function, methodically changing values in the function until the axis labels came within the range of the traditional alphabet. Others consistently sought out single, two, and three- letter words, as well as mappings for the most common letter in the English language ("e") as a path for breaking codes. We have discussed incorporating a prediction capability to the software, which could alert the teacher or team itself when a group is in danger of depending solely on a particular pattern of code breaking.

3.3.2. The choice for group over individual. Our design decision was to bias the function of the system to support groups over individual students. This was a choice we made early on in order to be able to experiment with the capabilities of the teacher station data collection. This decision proved to be extremely constraining in the classroom, and needed many socially engineered workarounds. Students wanted to play and explore when they had PDAs in their hands, but were unable to do so when it wasn't their day to communicate for their group. They had trouble orienting to these rules, and often competed for control of the stylus and PDA that would communicate with the server. Many cursor wars broke out in groups. When the wars were resolved in favor of strict, daily roles, some students became alienated. They were discouraged, and in many instances bored. This led them to discover other features of the PDAs such as appointment books, animated messages, and games such as Solitaire. Our desire to have the group at work and record all group transactions resulted in some students tuning out when they had a less-than-critical role assigned on a day. Schools also rely on individual accountability, and the system was unable to account for individual problem solving activity. As a workaround, we sought to set up each student as his/her own group, but the system could not handle 30 groups at a time.

3.3.3. Rethinking record keeping functions. When we designed *Code It!* we were pleased to experiment with the kinds of information and data that the server could provide to the teacher. The server did keep track of a great deal of information about group problem solving. When a group would solve for a code, the server would congratulate them and create a record of the solution. Unfortunately, these records were not immediately recoverable by the teachers and more work will have to be completed to develop useable data on student work. We discovered that students also needed access to records of their transactions and solutions. It became a practice for students from groups to share the ways they approached and broke codes. Even though groups kept notebook records of their solution strategies, students found it difficult to present their strategies after the fact.

We are hoping that in the next version of the software we can support easy to implement access to records of problem solving by the students. We seek to address the ideal balance between scaffolding for collaboration and individual exploration of the encoded functions, within the *Code It!* software environment, so as to lessen opportunities for disruptive stylus-wars. Our upcoming iteration will incorporate a flexible grouping mechanism, while maintaining both the appeal of collaboration and software robustness that made this first experience with *Code It!* successful. We are also considering ways to decouple the individual PDAs from the system. This would allow for students to explore and problem-solve individually as well as in groups, and as well as provide opportunities for "anytime, anywhere computing", home and family connections.

4. Conclusion

We had a successful field test of the *Code It!* wireless application under extremely constrained, yet real, classroom conditions. Now that we have an idea of how students and teachers actually put such software to use, we can develop a next version based on these interactions. This will result in a more aesthetically pleasing and useful application that fits the interaction, content, and assessment demands of the middle mathematics classroom.

5. References

1. *Middle-school Mathematics through Application Project Final Report to the NSF* (1998). Menlo Park, CA: Institute for Research on Learning.

2. Roschelle, J., Pea, R., Hoadley, C., Gordin, D., & Means, B. (2001). Changing how and what children learn in school with collaborative cognitive technologies. In M. Shields (Ed.), The Future of Children (Special issue on Children and Computer Technology, published by the David and Lucille Packard Foundation, Los Altos, CA), Volume 10, Issue 2, pp. 76-101.

3. Goldman, S. (2001). Technology in the Mathematics Classroom: Guidelines from the Field. *ERIC Update. ERIC Clearinghouse on Information and Technology*, 22 (2). (http://ericit.org/newsletter/Volume22-2/goldman.shtml)

4. Means, B., Penuel, W. R., & Padilla, C. (2001). *The Connected School: Technology and Learning in High School.* San Francisco, CA: Jossey Bass.

5. Soloway, E., Grant, W., Tinker, R., Roschelle, J., Mills, M., Resnick, M., Berg, R., & Eisenberg, M. (1999). Science in the palm of their hands. *Communications of the ACM, 42*(8), 21-26.

6. Curtis, M., Luchini, K., Bobrowsky, W., Quintana, C., & Soloway, E. (2002). Handheld Use in K-12: A Descriptive Account. *Proceedings of The First IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'02, pp.* 22-30*)*. New York: IEEE Press. .

7. Luchini, K., Bobrowsky, W., Curtis, M., Quintana, C., & Soloway, E. (2002). Supporting learning in context: Extending learner-centered design to the development of handheld educational software. In *Proceedings of The First IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'02, pp.* 107-111*)*. New York: IEEE Press.

8. Vahey, P., & Crawford, V. (2002, September). *Palm Education Pioneers Program: Final Evaluation Report.* Menlo Park, CA: SRI International. (http://www.palmgrants.sri.com).

9. Roschelle, J., Patton, C., & Pea, R. (2003). To unlock the learning value of wireless mobile devices, understand coupling. In *Proceedings of The First IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'02, pp. 2-7)*. New York: IEEE Press.

10. Tatar, D., Roschelle, J., Vahey, P., & Penuel, W. R. (in press). Handhelds go to school. *IEEE Computer*.

11. International Technology Education Association (2000). *Standards for Technological Literacy: Content for the Study of Technology.* Reston, VA: ITEA.

12. President's Committee of Advisors on Science and Technology and Panel on Educational Technology (1997, March). *Report to the President on the Use of Technology to Strengthen K-12 Education in the United States.*

13. Singh, S. (2000). *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography.* New York: Anchor Books.

14 Lichtenstein, G. Weisglass, J., & Erickan-Alper, K. (1998). *Final Evaluation Report: Middle-school Mathematics through Applications Project.* Denver, CO: Quality Evaluation Design.

15. Brown, A. (1992). Design Experiments: Theoretical and Methodological Challenges in Creating Complex Interventions in Classroom Settings. *Journal of the Learning Sciences*, *2*(2). 141-178.

16. Cobb, P., Confrey, J., diSessa, A., Lehrer, R., & Schauble, L. (2003). Design Experiments in Educational Research. *Educational Researcher, 32* (1), 9-13.

17. Design-Based Research Collective. (2003). Design-based research: An emerging paradigm for educational inquiry. *Educational Researcher, 32* (1), 5-8.