An electronic encounter log’s failure to scale
Walton Sumner, MD; Phil Asaro, MD
Washington University School of Medicine

ABSTRACT
We have developed a series of Electronic Student Encounter Log (ESEL) programs intended to introduce medical students to promising medical informatics concepts. We attempted to expand ESEL’s scope from ambulatory settings to all hospital venues and to track progress toward educational goals. Students were confused and frustrated by a previously fast interface and delayed feedback. Numerous scaling problems emerged. Our attempt to address these problems in an extensive revision, developed for the latest affordable hardware and operating system, failed due to new data-corrupting crashes. Risks of scaling up and other familiar software development lessons are reinforced.

INTRODUCTION
Clinicians are frequently ambivalent about electronic health record (EHR) systems. Data entry difficulties and infrequent opportunities to benefit from recorded data constrain enthusiasm for the benefits of amassing machine-readable medical records. Residents at our institution have complained that the ambulatory EHR “is never going to help me”.

While we cannot change vendors’ systems, many schools use handheld computers in medical education.1 For several years we have developed Electronic Student Encounter Log (ESEL) programs that record a small amount of information about students’ clinical encounters.2 Previously reported ESEL programs recorded age, sex, race, and diagnoses.

In spite of their modest contents, ESEL programs deployed in ambulatory settings performed various interesting functions. First, ESEL programs immediately notified a user recording a reportable disease, adapting the notification to the state where the student was working. Second, ESEL programs scanned lists of problems to produce demographically and clinically appropriate ‘ideas’ which the student could review.3 Ideas included alerts regarding likely but missing diagnoses, contact information for relevant trials, opportunities to treat two or more problems with one intervention, and warnings that likely treatments for one disease risk exacerbating a second disease. The idea feature was most valuable when the student recorded multiple problems. Third, ESEL data could be harvested, analyzed, and used for guidance, feedback, and even grading in primary care settings. Rapidly accumulating data allowed us to advise students of the problems they would likely encounter with specific preceptors. This was very interesting to MD-PhD students beginning their clinical training in family medicine. Students usually ignored problems such as smoking, other substance abuse, and depression until we developed feedback mechanisms and statistical grading processes that rewarded attention to these problems. ESEL facilitated recording of primary care problems, eventually supporting three distinct data entry preferences. ESEL’s most highly tailored data entry screens required a single tap to record a disease, and were arguably faster than paper for recording typical primary care encounters.4 After developing these features in ambulatory settings, we hoped to introduce students to similar capabilities in all third year rotations. Our first attempts failed.

METHODS

ESEL 5, Base program
ESEL 5 was a stable program constructed with Satellite Forms 5, requiring 10 to 20% of one developer’s time over 5 years, and tailored to primary care settings and Palm Tungsten C computers (400 MHz, physical keyboard). The program was deployed by loaning a handheld to a student, or beaming a compressed file to unpack on a student handheld. Similar files replaced defective tables or updated an ESEL program while leaving patient data tables unchanged.

Students used ESEL 5 in two settings, first an ambulatory 4-week rotation in Internal or Family Medicine, and second in a 6-week Pediatrics rotation. Students specified their course, then could add one of five course-specific new patient types. In Family Medicine the new patient types were distinguished by age: Adult, Teen, Child, Infant, and Baby. In other courses new patients were distinguished by organ system or complaint. Students recorded 70% of diagnoses from new patient screens.4
Patient demographic data included a free text identifier, age, sex, race, and date of visit. Free text was encrypted when not on display. Common problems were recorded by tapping a button labeled with an abbreviation (e.g. HTN). These buttons had checkbox behavior, and each called a global subroutine to create or modify a problem record linked to the current patient record. The subroutine checked new problems for mandatory reporting requirements.

On opening a patient summary screen, a series of global subroutines comprising 50,000 characters of code assessed whether any “ideas” were relevant to the patient. This algorithm performed integer math on the products of prime numbers assigned to recorded problems and similarly derived products assigned to ideas. This system detected missing diagnoses without performing sorts and string comparisons.

ESEL maintains records of all screens viewed, and whether it has closed gracefully or not. Following a crash, ESEL would ask what the student was doing when the crash occurred. We corrected all known causes of crashes in ESEL 5.

Students used infrared beaming to transmit 11 ESEL 5 tables to an administrative handheld. Beamed records from multiple students were appended to 11 files. A collector program, also written in Satellite Forms, imported these data and partially copied them, discarding sensitive or unnecessary information. The copied data were synchronized in a batch to the administrator’s computer. This allowed students to use their handhelds without needing to synchronize with the administrator’s computer, and circumvented the absence of wireless connections. One person (WS) collected all data for Internal and Family Medicine. An administrative assistant collected data for Pediatrics.

An administrative analytic program summarized data from up to 12 Internal or Family Medicine students, and plotted quality measures for each one using student-specified nicknames. Students could receive graphic, comparative feedback within 30 minutes of gathering for class.

**ESEL 6, Expanded program**

ESEL 6 was adapted from ESEL 5 in Satellite Forms 6 by one full time developer equivalent for 3 months in the spring of 2006, and then supported by 20% effort from each of us. Students were offered Palm Tungsten E2 computers (200 MHz, grafiti-2 handwriting recognition), although some invested in LifeDrive, Tungsten T5, Zire, or other Palm models.

Student identification across courses was attempted using unique identification numbers issued by the registrar and recorded by the student. Obstetrics, Surgery, Neurology, and Psychiatry courses were added. Each course required five new patient screens, each of which contained 20 to 30 diagnosis buttons. Shared resources, such as a sorted 1,200 item diagnosis list, were expanded. Screens recording procedures and visits were added and enhanced, with attention to documenting course requirements.

Data were again beamed from student handhelds to course administrator’s handhelds, and then imported into a collection program. A conduit program transmitted data from collection programs through firewall-protected desktop computers to a central SQL Server database. Students could identify and erase batches of records that had been beamed from their handhelds.

We defined course goals using lists of criteria stored on the server. We eventually developed SQL procedures to clean student data and determine whether the student had met a particular course goal. We used Crystal Reports to produce summary reports of student performance for administrators through a secure web interface.

**ESEL 7, Revision of the expanded program**

ESEL 7 was adapted from ESEL 6 in Satellite Forms 7 by one full time developer equivalent for 3 months in the spring of 2007, with review by an advisory group. Third party extensions to Satellite Forms were used to scroll lists, perform fast string comparisons and linear searches through tables, and erase temporary tables. All students were given Palm Tungsten T5 computers (200 MHz, 64 MB RAM, dynamic input area for data display or text input at programmer’s direction). Initial testing occurred on Tungsten C and E2 handhelds.

A small risk to protected health information remained in ESEL 6. ESEL 7 lengthened the series of malicious actions, beginning with a timely theft, required to recover any free text field.

Problem selection interfaces were converted to dynamically generated lists rather than static arrays of buttons. Lists were adaptive to patient
age, sex, and race. Sorting by ICD code, which groups many related items in lists, was considered intolerable. Thousands of 25 character problem descriptions were reviewed and rewritten to sort alphabetically to predictable places in a list.

Recording separate visits during a hospital stay was considered a confusing distraction from the task of tracking progress toward educational goals. Records intended to describe, for instance, the date and supervisor who monitored the student’s preoperative visit, were generated with minimal content and placed well outside of the standard documentation path. The idea reminder system was restored. We added a simpler, faster, but less versatile “hint” reminder system based on comparisons of 1 to 3 sorted ICD 9 codes. We added an encrypted “To Do” list.

A list of course goals was added to the program, and algorithms were developed to track progress towards completion of each goal. Buttons to adjust progress counts, allowing a student to manually reconcile ESEL 7 records with SQL generated reports, were considered dangerous and removed. In their place, a four table join was implemented with a new table and procedural code to dynamically recalculate progress after changes in patient and problem records, and a table was added to maintain memory of patient records that had been beamed to a collector program and deleted from the handheld.

RESULTS

Compiler did not scale
Satellite Forms 6 compiled early drafts of ESEL 6 in about 30 minutes. A series of experiments demonstrated that each reference to a global subroutine caused Satellite Forms to recompile all global subroutines from scratch, and later to write the many identical results to disk. Adding new patient screens with hundreds of global subroutine calls caused compilation times to increase exponentially and usually fail. The reminder system was the largest non-critical subsystem in the global code, so it was scrapped in ESEL 6. The problem was reported to Thacker Network Technologies, who repaired it in Satellite Forms 7, released in 2007. With the reminder system added back, Satellite Forms 7 compiled ESEL 6 and ESEL 7 in seconds.

Data losses following program update
We discovered a compelling need for a table structure change to support expanding goal tracking needs shortly after ESEL 6 was deployed. The table described events taking place during visits, such as specific physical examinations. A comprehensive update containing minor bug fixes, small enhancements, and the table change was distributed by beaming and email. Administrative data collection programs were updated simultaneously in a coordinated effort during a natural break in the course schedule. Students frequently elected not to install the update “because the program was working,” or did not run the update program after installing it on the handheld. Failing to anticipate the persistence of the old table format, the collection programs began to receive visit table data in mixed formats. This corrupted the collection programs’ visit data table, resulting in loss of all visit data from students who beamed records at the same time as an outdated program. The loss was immediately apparent on connecting to the server, but which handheld was at fault was less obvious. Eventually we located all outdated programs using server records, but some visit data were lost haphazardly for several months.

Student identity problems
Students could enter and edit their ID number. Many students did not know this number, and recorded fabricated numbers or social security numbers. Sometimes students changed their number. While not affecting the handheld, this number was the primary key linking all data on the server. Some handhelds were set to incorrect dates, for instance a year prior to the actual date. Following these problems we lost confidence that the course would be correctly set.

Main interface did not scale
The previously successful user interface failed in two ways. First, while ESEL 5 users usually recorded patient encounters on paper, and had to learn one PDA interface for a four week rotation, and another for a 6-week rotation, ESEL 6 users had to learn a new, complex set of button arrays every 4 to 6 weeks. Students frequently complained that they did not know how to find concepts, including concepts that were prominently displayed but abbreviated to fit on a small button. Course masters worried that students spent too much time trying to document patient encounters. Second, while ESEL 5 users were asked to document reality in anticipation of receiving decision support and statistical review, ESEL 6 users were asked to document progress toward a list of goals, with little feedback. Although the new patient screens could record di-
agnoses related to goals easily, the actual goals were not listed, the web-based process for displaying goal progress took longer than expected to implement, and when a goal was not met the student had no explicit instruction on how to record progress toward it. This was especially true for procedures, even after several revisions in procedure recording screens.

Risks scaled up, incentives did not
Without the reminder system or timely feedback, ESEL 6 lost documentation incentives. We expected that drug and disease reference software would compel students to carry their handhelds, but this was not true when students wore scrubs. Palm computers do not fit snugly in any scrub pocket, and students frequently shed their white coats. As a result, Surgery and Obstetrics and Gynecology students constantly worried about dropping or losing their handhelds. The E2 did not serve any critical function, such as telecommunication, so students in scrubs stopped carrying their handhelds, and soon left them at home.

Data recovery overload
An as yet unexplained problem occurred while recovering patient problem data from multiple students using ESEL 6. Students record multiple problems per patient, so that problem tables grow faster than any other patient data table. Students seldom removed old records, adding to the volume. The administrative data collection program could not complete the partial copying of large numbers of problem records in preparation for synchronization with the SQL Server database, although ample memory was available to copy the desired data. Unfortunately, the collector program could corrupt an entire batch of problem data. We quickly learned not to collect more than 3 or 4 students in a single batch. We eventually implemented an empiric copying limit of 4,000 problem records. After copying 4,000 records, we could synchronize the collector with the SQL Server database, erase the synchronized tables, and copy the next 4,000 problem records. Many problem records were lost while we learned to circumvent this limitation.

Delayed feedback
Neither students nor preceptors had access to progress reports until the third quarter of the academic year, when we finally delivered a web-based report generator for administrators. The initial reports did not reflect some students’ expectations regarding their progress toward goals. Given the problems described above, all courses reverted to paper records in the spring of 2007, and we began rebuilding ESEL-7.

Hardware problems
Many students bought handhelds before the decision by central administration to loan Tungsten E2 handhelds with ESEL 6 to all students. Only a few Tungsten E2 units failed out of 100 deployed. Beaming from Tungsten C, E, E2, T5, and Zire models was generally reliable, although hardware problems occasionally disabled infrared beaming completely. Beaming from two students’ Life Drives was always difficult and sometimes could not be accomplished; Treo’s were only slightly more consistent. Beaming was never graceful for groups of more than 10 students gathering for lectures, even if the handhelds worked perfectly. Procedures to recover data using SD cards or table backups on home computers had not been developed.

Although Tungsten C handhelds were consistently stable, certain combinations of third party extensions to Satellite Forms and specific hardware models, notably the E2, crashed ESEL, limiting some development options in ESEL 6.

ESEL programs were not stable on the Tungsten T5 model. One student reported problems using ESEL 6 on a Tungsten T5, but attributed these problems to a general lack of technological savvy. Not recognizing this warning sign, we purchased 150 Tungsten T5 handhelds to deploy with ESEL 7. We expected the T5’s high clock speed, dynamic input area, and non-volatile memory to justify the added cost relative to the E2. A number of situations caused ESEL 7 to crash with a cache error on the T5. Some crashes involved boundary conditions and third party extensions to Satellite Forms. For instance, an extension might crash if called to manipulate an empty table. The extensions were practically indispensable from a performance perspective, so we focused on detecting boundary conditions to avoid problematic calls. Unfortunately, the cache errors were malignant: after suffering one crash on a T5, especially if a cache error occurred, ESEL 7 was unstable and crashed at progressively earlier steps, quickly becoming useless. ESEL 7’s extensive security, list management, goal tracking, and decision support revisions needed to be delivered completely bug free. The greater stability of ESEL on the Tungsten C and E2 delayed discovery of platform dependent bugs and the cache error problem. One week before deployment we discovered and
corrected multiple sources of cache errors. One day before deployment we discovered more, which we were not able to correct in the ensuing week. Our Palm-based project was suspended. Nine months later Satellite Forms provided a patch to circumvent a “busy bit” that the T5 operating system was not clearing. The patch prevents all ESEL 7 cache errors, but too late.

DISCUSSION

Physicians constantly extrapolate from personal memories, anecdotes, and narrowly constrained trial results to care for the next patient and set policies. Meanwhile, although we have a scientific bent and training comparable in length and intensity to a doctor of philosophy, we capture mere droplets from the river of data that daily passes us. It is inconceivable that a successful PhD candidate in the biological sciences could record so little about their experiences in the laboratory. The value of capturing clinical data consistently is immense, yet our policies encourage us to record just enough to satisfy external mandates. These policies include educational goal tracking in medical school, counting procedures in residency, enumerating billable diagnoses and interventions in practice, and tracking quality measures in hospitals and groups. These activities would be good even if we were not paid to do them, but they lose some value when they distract us from documenting patient experiences completely enough to learn from the past. Unfortunately, it is hard for us to show our students examples of the value of documentation in clinical settings – no local software really delivers on the promise. ESEL should provide one example, albeit limited in scope. A thorough data capture device, as student encounter logs go, should deliver more numerous and relevant insights as documentation improves. Sadly, ESEL-6 and 7 exemplified only the pitfalls of medical informatics, in surprising confluence.

Lessons

We experienced numerous setbacks related to hardware, software, user interface, expectations and communications. Some of these are:

- Version control, identity, and data disposal problems accompany a shift from course-owned handhelds to student-owned handhelds.
- A button array interface is frustrating, not fast, if the user has to learn new arrays frequently. Stable button arrays might work very well for residents and practicing physicians.
- Infrared beaming is not a practical way to recover data from many users: some form of wireless connection is almost imperative. Mobile device software designers should plan redundant data retrieval procedures lest the primary mode fails.
- Students who are familiar with new consumer technologies may be unaware of basic informatics concepts.
- Rapid application development environments, third party extensions, and operating systems can interact badly. Early, aggressive testing on the target hardware and operating system is a reasonable demand.
- Server tasks should be performed on servers, and SQL tasks in SQL. Replicating SQL joins in procedural code on a handheld is possible, but time consuming to write or execute, and fraught with hazard.
- Requests for security, alphabetization, tracking, interface changes, etc, have large but predictable implications that requesting parties need to hear and acknowledge. The development team must communicate these facts.
- Tension between different participant’s goals – in our case between goal tracking and thorough documentation – needs to be reconciled or managed proactively.

We deployed the 2008 version of ESEL, written in Visual Basic for Windows Mobile 6 in June 2008. This much simpler program tracks goals in minimal detail, but synchronizes on a wireless network and had Internet reporting available immediately. About 2 programmer-years were committed to the effort. We hope to restore patient-centered data entry, dynamic problem lists, and decision support features in future versions.

Bibliography