Usability Evaluation of the Progress Note Construction Set

Steven H. Brown, MS MD[1,2], Shawn Hardenbrook MSW[1], Linda Herrick RN PhD[3], Judith St. Onge RN PhD[1], Kent Bailey[3], Peter L. Elkin MD[3]. VA Tennessee Valley Healthcare System[1], Vanderbilt University[2], and the Mayo Clinic[3]

Overview: The Veterans Administration (VA) Computerized Patient Record System (CPRS) is a nationally deployed software product that integrates provider order entry, progress notes, vitals, consults, discharge summaries, problem lists, medications, labs, radiology, transcribed documents, study reports, and clinical reminders. Users rapidly adopted the graphical user interface for data retrieval, but demanded options to typing for data entry. We programmed "point and click" forms that integrate with CPRS individually, but were soon overwhelmed by requests. Subsequently, we developed the Progress Note Construction Set (PNCS); a tool suite that permits subject matter experts without programming skills to create reusable "point and click" forms. In this study, we evaluate the usability of these user-constructed forms.

Methods: An untrained, non-VA subject matter expert used the PNCS to create a graphical form for "skin tear" documentation. Ten VA nurses used the skin tear form to document findings for 7 standardized clinical scenarios. Following each scenario the subjects answered usability questions about the form.

Results: The subject matter expert created the skin tear form in 78 minutes. Users found the form to facilitate their data entry (p 0.0265), and to be at least as fast (p 0.0029) and as easy to use as expected (p 0.0166). Average note entry time was 3.4 minutes.

Conclusion: The PNCS allowed a non-programmer to quickly create a usable, CPRS-integrated point and click form. Users found the subject matter expert's form fast and easy to use. The tool suite is a more scaleable form creation method because capacity is no longer limited by programmer availability.

Introduction

Computer-based medical records improve access to patient-specific information via multi-user sharing, automated search functions, legible text, and consistent structure [1-8]. To realize the information access benefits of computer-based medical records, systems designers, implementers, and end-users must address the challenge of data entry. Some systems adopt a strategy of gathering only clinical information that is easily available in electronic format, such as labs and transcribed reports. Less commonly, systems have been designed to facilitate direct provider entry of clinical information via innovative combinations of structured data, free-text, and occasionally even multimedia format[9].

Despite the challenges, computer-based medical records are being widely implemented. The Department of Veterans Affairs[10-13], the Department of Defense[14], Kaiser-Permanente and numerous other organizations have made a commitment to computer-based patient record systems [2, 15-21] in hopes of improving information access.

The VA's "Computerized Patient Record System" (CPRS) is deployed at each of its 173 Medical Center campuses. Using a tabbed chart metaphor, CPRS integrates provider order entry, progress notes, vitals, consults, discharge summaries, problem lists, medications, labs, radiology, transcribed documents, and study reports. Providers use CPRS to enter, edit, and electronically sign documents. At the VA Medical Center in Nashville (recently merged and renamed the VA Tennessee Valley Healthcare System) we began preparing for CPRS in 1997 and went "live" in 1998[12]. End users' reaction to CPRS's patient-centered organization and graphical user interface for data retrieval has been overwhelmingly positive. We have encountered less enthusiasm for direct provider data entry. Specifically, many providers have resisted typing their notes.

Our initial response to diminish the data entry burden was to use a nationally distributed template-building tool, the Text Integration Utility (TIU), to customize input screens. Templates are said to improve documentation consistency and quality[22, 23]. TIU provides tools to define and name templates, to store and retrieve documents, to track document status, and to manage business rules including the electronic signature process. A TIU template consists of pre-formatted text, and placeholders for patient-specific data such as the most recent chest x-ray. TIU templates are completely text-based, and lack common graphical widgets such as radio buttons and drop-down boxes. This design choice insures compatibility with a large installed base of character-only terminals. The providers' response to our TIU

1067—5027/01/$5.00

template effort was less favorable than we had hoped. In addition to the usually cited complaints (e.g. "I didn't go to medical school to be a clerk"), we also commonly heard that TIU's text-based data entry facilities are difficult to use when compared with other features of the graphical CPRS client software.

We subsequently embarked on a project to make graphical "point and click" templates available to our users. Templates were constructed using Delphi™, and integrated with our facility's production CPRS system. At present we have 54 unique Delphi-based forms that have been used in production 41,278 times. Each simple template takes approximately four hours for an experienced programmer to develop. Complex templates take several days. The forms were well received, and we were quickly overwhelmed with programming requests. Despite our best efforts, the template programming work queue grew continually.

In order to keep pace with demand, we decided to alter our strategy. We redirected programmer effort from individual form development to the creation of form building tools. The design goal of these tools, named the Progress Note Construction Set (PNCS), is to enable subject matter experts without programming skills to create point and click forms for both data entry and retrieval. In this paper we describe the PNCS, and evaluate the usability [24] of a graphical form created with it by a subject matter expert.

Methods

The Progress Note Construction Set

The PNCS is software developed at the VA Tennessee Valley Healthcare System (TVHS) that is used to make "point-and-click" forms for healthcare documentation that integrate with CPRS. PNCS allows non-programmers to graphically construct new templates by reusing components from existing templates, or by creating new ones. The template creator simply has to "drag" the component from a palette and "drop" it on the new form. PNCS takes care of the underlying details, such as hospital information system (Veterans Information System Technology Architecture, a.k.a. VistA) communication, report generation, business rules implementation, and electronic signature code validation.

PNCS consists of an object repository, an object browser, an object editor, an object interaction scripting language, a form design application, and a form run-time module for the client pc. Examples of form objects include a "shortness of breath" yes-no check box object, and a "chief complaint" memo object. Objects are categorized as "Simple" or "Complex". "Simple" objects are single Windows widgets (e.g. a check box) designed to collect a specific piece of information. Complex objects are named groupings of simple objects, such as "chest exam". Objects are created using the object editor, and stored in the object repository. Objects can be linked to the VistA database with the object scripting language so that at runtime patient-specific data, such as demographics and recent test results, are merged into the form.

The Form Design Application includes tabs for form layout, report design, script composition, and testing. It produces three definition files for each template: a form layout file, a report definition file and a script definition file. The form developer creates a form layout by dragging objects from the object browser and placing them on the evolving form. If new objects are needed, the object editor can be run concurrently with the form designer.

Report scripts specify how to transform patient-specific data gathered during form use into a textual report that is subsequently uploaded into CPRS. Report scripts can be used to define formatting, to specify phrases to be merged with patient data elements (e.g. "the patient reports x"), and to indicate if data elements left blank by the note author are to be included. The latter feature addresses the common complaint that TIU templates produce lengthy, mostly "blank" notes. The object scripting language allows simple interactions between objects such as "if the answer to question X is "yes" then display object Y", or "enter the sum of the numbers in objects X, Y, and Z into object A.

The Form Design Application allows the user to test their form off-line before uploading it into the production environment. Form layout and script-defined behavior can be reviewed and modified. The test environment does not connect to VistA, so data merges are not evaluated in the test environment. Once designed and debugged, template definitions are stored on a read-only network directory so they may used in production.

The PNCS runtime module runs on the end user's workstation. The runtime module interprets the form definition files stored in a network directory and reproduces the graphical form for the end user. This approach allows form updates to the "live" production system without downtime. The runtime

controls data communication with VistA, such as saving and loading forms, and merging VistA data (e.g. lab values). The runtime module allows data entered into a PNCS form to be retrieved as a form (i.e. structured data) or as a text-based report. The runtime module uses the report definition script when the PNCS form is signed to generate text-based reports. The reports are then uploaded into TIU, with other non-PNCS clinical notes. This ensures that all notes may be reviewed in an integrated manner within CPRS.

Usability Evaluation of PNCS Forms

PNCS form usability was evaluated in two-steps. First, an untrained user with no VA or CPRS exposure created a new PNCS form. The test form developer was given a copy of a paper form request submitted by TVHS nursing staff. The resulting form is shown in figure 1.

The newly created form was imported into the TVHS production system and used by 10 VA nursing staff to enter documentation on 7 standardized clinical scenarios. The subjects received no instruction in how to use the form. After entry of each PNCS note, the study subjects answered 5 usability questions (figure 2). Note entry time for each scenario was recorded. The responses on the survey were coded and tabulated. Mean, standard deviation, and 95% confidence intervals were calculated. The Beta Binomial test was used for each measure to provide a population result estimate and to determine the finding's significance (i.e. the likelihood that the results could have been arrived at by chance alone).The Beta Binomial test does not assume independence within subject.

Results

An untrained subject matter expert created the form used in the evaluation in 78 minutes (figure 1). The average age of the 10 respondents who subsequently used the form to document the experimental clinical scenarios was 45 years. All respondents were RN's, with an average of 18 years in nursing and 12 years of VA service. Nine respondents were female. Four respondents were from the Nashville campus and 6 were from the Murfreesboro campus. Six respondents were daily CPRS users, each of the other categories (weekly, monthly, less than monthly, and never) were represented once. Six respondents were daily personal computer users, only one had never used a personal computer before.

Usability question response means and 95% confidence intervals for are shown in Table 1. The mean time to complete the each task was 3.4 minutes. Beta binomial test (BBT) results are shown in table 2. For questions 1,3,4 and 5 the p values indicate that the study results should not be expected by chance alone. Most users were able to completely or mostly enter the assessment data (p 0.09 BBT). Data entry was as easy or easier than expected ($p < 0.0166$, BBT) and as fast or faster than expected (p 0.0029, BBT). Most users felt that the screen design facilitated their data entry (p 0.0265, BBT). Most user suggestions (24/25) asked for additional objects to use for detailed documentation (e.g. a space for comments). One comment noted a minor TIU

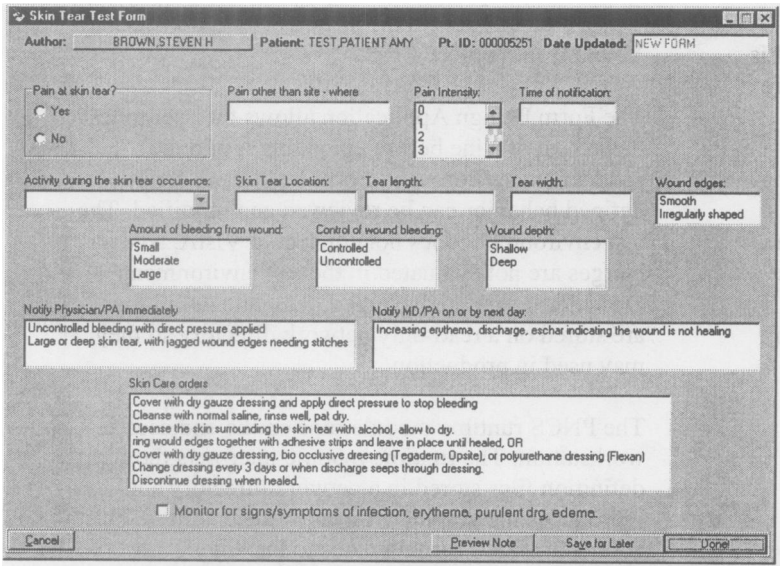

Figure 1. PNCS Skin tear form created by subject matter expert.

1. Were you able to enter the assessment data?
   Completely, Mostly , Somewhat, Minimally, Not at all.

2. Did the data entry options meet your needs?
   Completely, Mostly , Somewhat, Minimally, Not at all.

3. Was the data entry
   Easier than expected,
   Just as easy as expected
   Harder than expected

4. Was the data entry
   Faster than expected
   Just as fast as expected
   Slower than expected

5. Did the screen design facilitate your data entry?
   Yes No

Figure 2. Usability questions asked of nurse-users.

	Able to enter data	Needs met by entry options	Ease of Entry	Speed of entry	Entry Facilitated	Completion Time
N	69	69	69	69	68	38
Mean	2.09	2.22	1.62	1.59	1.22	3.41
CI - 95%	+/-0.23	+/-0.24	+/-0.17	+/-0.17	+/-0.10	+/-0.69

Table 1. Mean and 95% confidence intervals of questionnaire responses. For questions 1 and 2 the most favorable possible answer was coded as 1 and the least favorable answer as 5. For questions 3 and 4 the coding range was 1 to 3. For question 5 the coding range was 1-2.

Question	Response	Mean Estimate	Significance	95% CI
1 - Able to enter data	Completely or Mostly	0.70	0.090	(0.47,0.87)
2 - Needs met by entry options	Completely or Mostly	0.64	0.30	(0.34,0.84)
3 - Ease of entry	Easier or as Easy	0.83	0.0166	(0.57,0.96)
4 - Speed of entry	Faster or as Fast	0.87	0.0029	(0.65,0.97)
5 - Entry facilitated	Yes	0.79	0.0265	(0.54,0.95)

Table 2. Results of the Beta Binomial test for usability questions.

configuration problem that was unrelated to PNCS.

Discussion

The Progress Note Construction Set was developed to help non-programmers create data entry forms that allow VA users to enter clinical documentation into CPRS. In this study we asked the question - could a novice developer create a data entry form that is usable for routine documentation? The answer to the first part of the question, can a novice use the PNCS tools to create a data entry form, is "yes". In 78 minutes a new user created a form for the documentation of skin tears from a submitted paper draft. In our experience, gained during the Delphi programming of 54 forms, the skin tear form would have required approximately 4 hours of programmer time to create. While this study did not focus on form creation usability, we were pleasantly surprised at the speed with which the study form was created using PNCS.

The answer to the second part of the question, might a form created by a novice actually be usable? is a qualified "Yes". For each scenario, the test subjects felt the form was quick and easy to use. Data entry needs were partially, but not completely, met. It should be noted that the tested form was a first effort, which had yet to undergo use-based revisions. Our experience with programmed forms is that 1-3 modification-feedback cycles are needed before a form can be considered finalized. Virtually all criticisms offered by the test subjects (e.g. "need a place for comments") stemmed from design oversights on the original paper request. Furthermore, each suggestion could be easily addressed with the PNCS tools without programmer intervention. The tools were never intended to eliminate the "up-front" analysis required to design a functional form, or to eliminate the need for user-feedback driven form improvements. In the current evaluation, the PNCS tools met the design requirement of enabling a non-programmer to create a useable "point and click" form.

The development of tools that allow non-programmers to create templates is not novel. Franklin et al. developed a tool set for the creation of order entry templates[25]. Our projects have significant similarities despite being developed independently. Overwhelming demand for programming services motivated each team's efforts. Each solution shares a similar architecture (e.g. an editor, a database, and a run time environment). Finally, each solution reduced new template creation time roughly four-fold.

The results of this study are critically important to our institution because it means that the point and click forms creation process can shift from a programmer-limited "cottage industry" to a more scaleable model driven by functional analysts. In the short time since the creation of the PNCS, we have three new form developers and 20 PNCS forms in production. This fact, combined with an apparent reduction in creation time cost per form, has greatly increased our ability to meet customer demand.

In this paper, we describe a Fifth generation language designed in an object-oriented framework to allow users to create and reuse screen and database objects without programmer assistance. Users can easily incorporate the objects into a screen capable of accessing and updating HIS databases. We believe

that this technology may enable more rapid and distributed development of end-user tools for Health Informatics. In future work, we will determine if the PNCS can be used beyond the VA TVHS. If the technology can be transplanted to other VA Medical Centers, its impact will be greatly increased. It is our experience that most VA's have functional analysts capable of using PNCS, but very few have Delphi programmers. Other avenues of future work include the development of tools and processes to share forms and objects between sites. Such a mechanism could dramatically reduce per-site forms related effort, and allow sites unwilling or unable to create novel forms to deliver point and click functionality to their users.

References

1. Barnett, G.O., Computers in Patient Care. New England Journal of Medicine, 1968. **279**(24): p. 1321-1327.2. Greenes, R.A., et al., Recording, retrieval and review of medical data by physician-computer interaction. New England Journal of Medicine, 1970. **282**(6): p. 307-15.

3. McDonald, C.J. and W.M. Tierney, Computer-stored medical records. Their future role in medical practice. Jama, 1988. **259**(23): p. 3433-40.

4. Dick, R.S. and E.B. Steen, eds. The computer-based patient record: an essential technology for health care. 1991, National Academy Press: Washington DC. 190.

5. Powsner, S.M., J.C. Wyatt, and P. Wright, Opportunities for and challenges of computerisation [see comments]. Lancet, 1998. **352**(9140): p. 1617-22.

6. Shortliffe, E.H., P.C. Tang, and D.E. Detmer, Patient records and computers [editorial] [see comments]. Ann Intern Med, 1991. **115**(12): p. 979-81.

7. Schneider, E.C., et al., Enhancing Performance Measurement. NCQA's road map for a health information network. Jama, 1999. **282**(12): p. 1184-90.

8. Elkin, P.L., et al., The role of compositionality in standardized problem list generation. Medinfo, 1998. **9**(Pt 1): p. 660-4.

9. Dayhoff, R.E., et al., Providing a complete online multimedia patient record. Proc AMIA Symp, 1999: p. 241-5.

10. Kolodner, R.M., ed. Computerizing large integrated health networks: the VA success. Computers in health care, ed. K.J. Hannah and M.J. Ball. 1997, Springer-Verlag: New York. 515.

11. Anderson, C.L. and K.C. Meldrum, The VA computerized Patient Record--a first look. Proc Annu Symp Comput Appl Med Care, 1994: p. 1048.

12. Brown, S.H., No free lunch: institutional preparations for computer-based patient records. Proc AMIA Symp, 1999: p. 486-90.

13. Payne, T.H., The transition to automated practitioner order entry in a teaching hospital: the VA Puget Sound experience. Proc AMIA Symp, 1999: p. 589-93.

14. Staggers, N., The vision for the Department of Defense's computer-based patient record. Mil Med, 2000. **165**(3): p. 180-5.

15. Pryor, T.A., et al., The HELP system. Journal of Medical Systems, 1983. **7**(2): p. 87-102.

16. Stead, W.W., W.E. Hammond, and M.J. Straube, A chartless record--is it adequate? Journal of Medical Systems, 1983. **7**(2): p. 103-9.

17. Barnett, G.O., The application of computer-based medical-record systems in ambulatory practice. New England Journal of Medicine, 1984. **310**(25): p. 1643-50.

18. Bleich, H.L., et al., Clinical Computing in a Teaching Hospital. New England Journal of Medicine, 1985. **312**(12): p. 756-764.

19. Scherrer, J.R., et al., An integrated hospital information system in Geneva. MD Computing, 1990. **7**(2): p. 81-9.

20. McDonald, C.J., et al., The Regenstrief Medical Record System: 20 years of experience in hospitals, clinics, and neighborhood health centers. MD Computing, 1992. **9**(4): p. 206-17.

21. Camp, H.N. Technical Challenges, past and future in implementing THERESA - a one million patient, one billion item computer based patient record and decision support system. in Health Care Information Infrastructure. 1995. Philadelphia PA: Society of Photo-Optical Instrumentation Engineers.

22. Crist-Grundman, D., et al., Evaluating the impact of structured text and templates in ambulatory nursing. Proceedings - the Annual Symposium on Computer Applications in Medical Care, 1995: p. 712-6.

23. Henry, S.B., J.A. Morris, and W.L. Holzemer, Using structured text and templates to capture health status outcomes in the electronic health record. Jt Comm J Qual Improv, 1997. **23**(12): p. 667-77.

24. Elkin, P.L., et al., Standardized problem list generation, utilizing the Mayo canonical vocabulary embedded within the Unified Medical Language System. Proc AMIA Annu Fall Symp, 1997: p. 500-4.

25. Franklin, M.J., et al., Modifiable templates facilitate customization of physician order entry. Proc AMIA Symp, 1998: p. 315-9.