Observing End-User Customisation of Electronic Patient Records

Cecily Morrison and Alan F. Blackwell

University of Cambridge, Computer Laboratory
JJ Thompson Avenue, Cambridge, UK, CB3 9EU
{Cecily.Morrison,Alan.Blackwell}@cl.cam.ac.uk

Abstract. The contemporary practice of medicine, which is concerned both with national standards of audit and innovation through local customisation, is a prime domain for end-user development. In this paper we describe four experiences of end-user development in this domain that offer interesting empirical examples. We look at existing practices through considering end-user customisation of paper charts (1), compare the end-user customisation facilities provided by two applications for electronic patient records (2), assess the structure of an actual end-user development using one of these (3), and propose a longitudinal study of end-user customisation building on this work (4).

Keywords: Patient records, Healthcare, End-user customization.

1 Introduction

Contemporary medical practice is fundamentally concerned with the definition and execution of standardised procedures. The creation of new standard procedures by individual hospitals or units, and the local refinement of existing procedures, is commonplace. Such local customization, even if minor, can be seen in a positive light, encouraging reflective professional practice [1], as well as innovation. Despite these potential benefits that procedural diversity carries, it causes problems for the application of information technology to clinical practice, which for economic reasons is often deployed across a large number of client institutions. This conflict is well illustrated by the significant difficulties encountered in the implementation of the British National Program for Information Technology, a very large scale deployment of standardised clinical administration software [2].

The combination of requirements, for both standardisation and customisation, means that Electronic Patient Record (EPR) systems are a natural target for end-user development or end-user customisation. (In this paper, we will refer to both as EUD.) Indeed, leading EPR products offer in addition to their standardised set of procedures and record formats, significant capabilities to support local end-user development. The already established medical practice of defining and refining procedures makes EUD application use in medical environments particularly interesting to study, as the practitioners are familiar with the process but not the technology. Such a situation provides the opportunity to assess customisation procedures without the use

V. Pipek et al. (Eds.): IS-EUD 2009, LNCS 5435, pp. 275–284, 2009.

of technology as well as a capability to home in on the problems stemming from the technology use, as we can assume that the process of negotiating customised procedures is already smoothly established.

This paper highlights a number of examples from our experiences with customisation procedures and EUD technology employed in healthcare environments. We begin by investigating the process of customisation without technology, by detailing how paper charts were developed in an intensive care unit (1). Moving on to look at customisation of technology, we present a comparison of the initial customisation procedures of two EPR systems, GE Healthcare *Centricity EMR* [3] and IMDsoft *Metavision* [4] (2). We then focus on a single system, Metavision, and explore how system structure affects end-user development (3). Finally, we change focus from the initial to the long-term practice of customisation, and propose how one might study long-term EUD usage in a medical environment (4). Despite the brevity of these examples, we aim to demonstrate in our conclusion that studies in the medical environment can offer insight into a wide variety of issues in the EUD application development process.

2 Customisation on Paper (Case 1)

As noted in the introduction, the creation and refinement of new procedures in medical contexts is both common and productive. Indeed it is considered an important skill for senior clinical practitioners and a way to provide innovative patient care. These procedures are traditionally deployed through the development of appropriate charts. Below we look at the customisation of paper charts found in an intensive care unit (ICU) of a cardiothoracic specialist, research-oriented hospital in the UK. We first detail a brief example of process and then discuss the end-result.

2.1 Customisation of the CCOC Paper Chart

A tremendous amount of data about a patient's state is collected from the many machines to which she is attached – heart rate, fluid balance, oxygen levels and blood results to name just a few. This data is organized in various charts, perhaps as many as 10, for use by different kinds of practitioners (e.g. nurses vs. doctors). In order to utilize such a wealth of information, the director of this ICU discouraged narrative observations on charts, in favour of formalised tabular formats that could be easily reviewed and compared. The most commonly used chart shared by all practitioners, the Critical Care Observation Chart (CCOC), became, as a result, formalised to an extent that it did not accommodate the more diverse aspects of patient care with which nurses are concerned.

In response, nurses developed the habit of turning the CCOC over, to write less structured observations on the back. However, as unstructured text presents problems in consistency and standard interpretation, the ICU nursing staff decided to define a standardised structure for nursing observations too. They revised the CCOC by printing another grid on the reverse side, providing another, but different, knowledge structure for use by nurses.

We might note two points from this example of the customisation process. First, the process of formalisation and categorisation is a feature of organisational data management, a point observed by Bowker and Star [5] in an analysis of case studies that included the formalisation of nursing practices. It is therefore a process that is well practised, even when customisation happens without any EUD technology, a point that will be discussed further in section three. Second, it demonstrates a reactive, incremental developmental process which suggests the importance of looking at long term usage of patterns of EUD application and not just the initial period of customisation, something that will be done in section four.

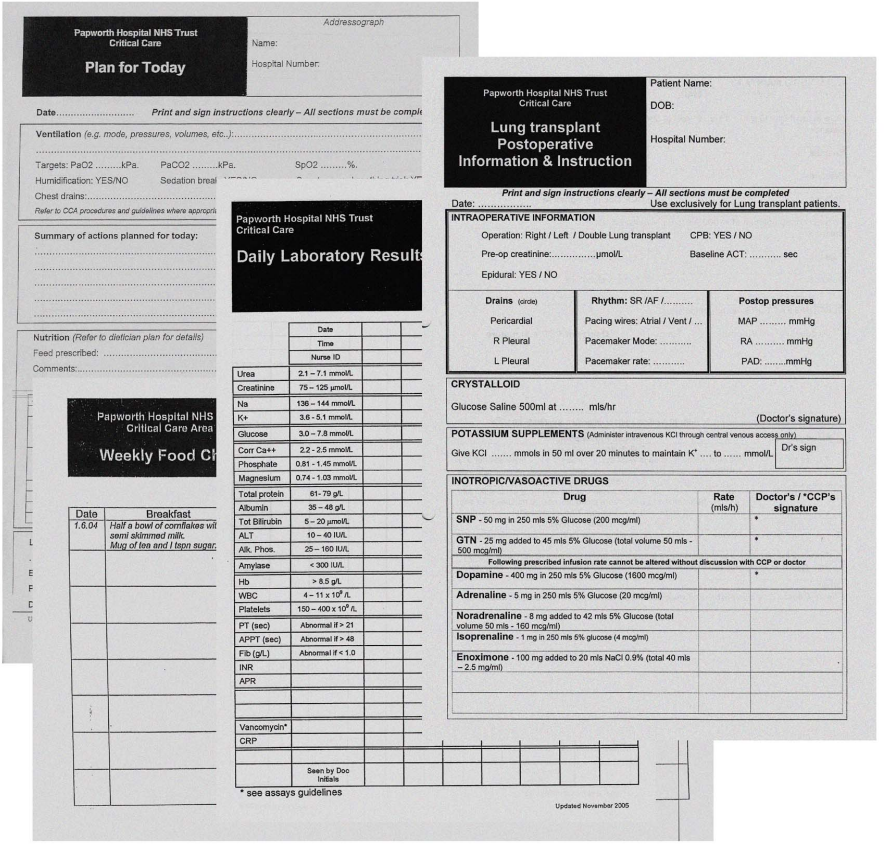

Fig. 1. A selection of more recently design paper charts in use at the ICU

2.2 Examples of Paper Chart Customisation

In Figure 1 and 2 we present a collection of charts from this ICU unit. Those in Figure 1 are newer and more standardized than those in Figure 2. In particular, we would like to draw attention to the visual coherence that begins to appear in the newer charts. This suggests that not only is customisation happening on a chart-by-chart basis as the

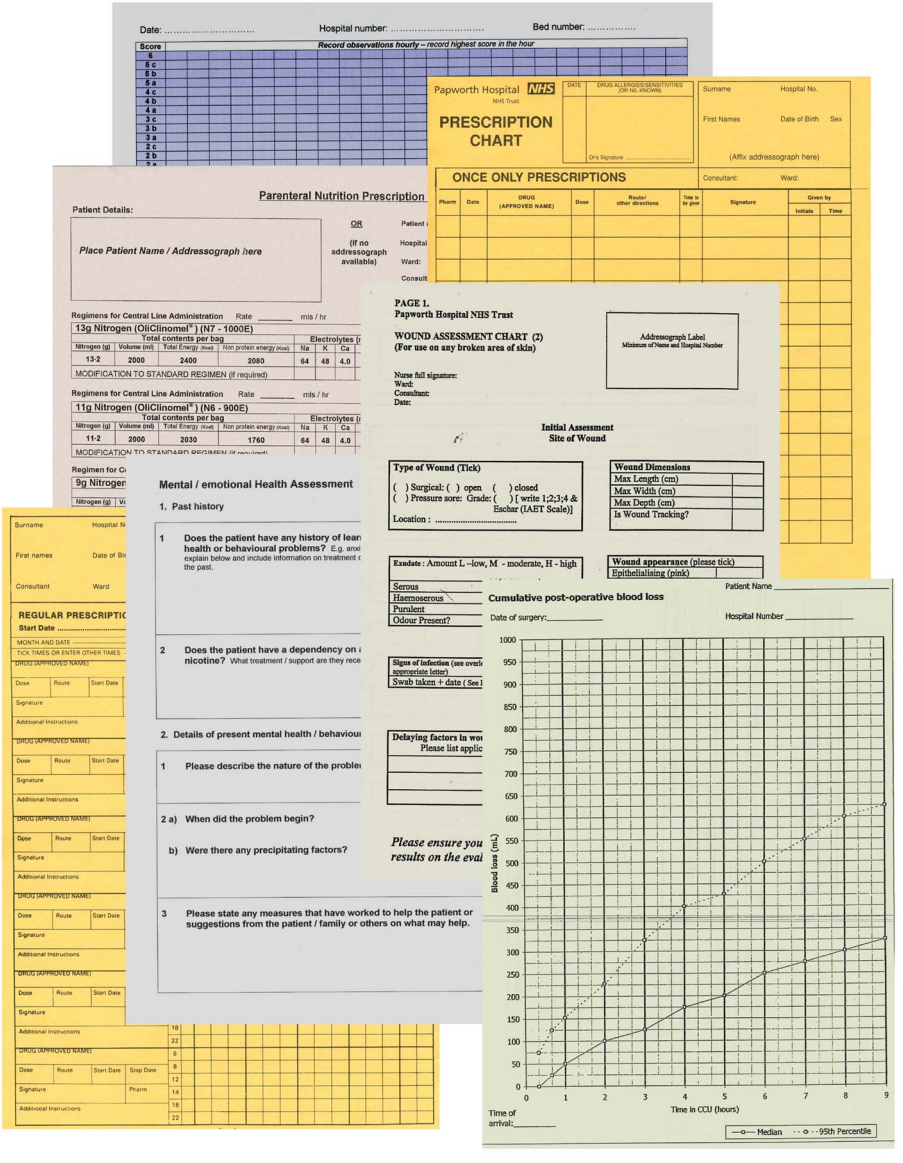

Fig. 2. A selection of older paper charts in use at the ICU

need arises (as above), but that there is concerted design of the information structure. Customising fields on a chart might be seen as end-user programming while design of an information structure may be likened to end-user software engineering. This distinction will be considered in more detail below.

3 Comparison of Two Health EUD Applications (Case 2)

3.1 GE Healthcare

The CHI 2006 workshop on End-User Software Engineering [6] included a representative of GE Healthcare, who described the customisation facilities of the Centricity Electronic Medical Record product [7]. The aspect of the system providing the focus of discussion was its facility for generating medical reports. Medical reports are standardised in structure and terminology, to an extent that suggests they might easily be generated automatically. Indeed, Centricity allowed doctors to define the content of an individual patient's report using menus and checkboxes, then automatically generate the full prose text report from the resulting data fields. However, despite the clear advantages of standardised text (consistency, quality control, efficiency), individual doctors often have strong preferences for particular expressions or writing approaches, to an extent that makes them reluctant to use standardised text.

The solution offered by the Centricity product is that doctors can customise report text generation in accordance with their own preferred style, using a "medical expression language." Unfortunately, the computational complexity of transforming a predefined set of terms into arbitrary prose constructs is such that this language includes most of the data and control features of a general purpose programming language (typed variables, conditionals, iteration and others). The instruction manual for the medical expression language resembled an introductory programming course, in both content and structure. Few doctors have the time to develop programming skills from scratch, leading on the one hand to a thriving third-party industry offering customisation services for Centricity, and on the other hand to a variety of costs and risks associated with the specification and debugging of any scripts written by relatively inexperienced doctors.

3.2 IMDsoft Metavision

The director of the above mentioned ICU considered the GE Healthcare product when implementing a new Electronic Patient Record system, but eventually selected the MetaVision product from IMDsoft. He reported to us, at the outset of our research, that customisation facilities had been a significant factor in that decision. Although MetaVision has a different set of customisable features, the encounter with a different product did provide us with an opportunity to compare end-user customisation and deployment issues associated with different products in the same market. At this ICU, report formats also provided an initial focus for customisation. However, it was not prose reports that were the main priority, but the layout and content of patient charts to monitor patient's vital signs, drug administration and fluid balance.

Implementation of Metavision at this ICU was mainly concerned with replicating the structure of these existing paper charts within the Electronic Patient Record. We observed the necessary customisation work being carried out by a small team of system "super-users" – the clinical director of the unit (a consultant anaesthetist) who had been the initiator and driver of new policy, the director of nursing, and the computer system administrator. Their approach to the project closely resembled professional practice. Having collected samples of all the paper charts used in this ICU, the

team identified all data fields. They arranged these on flipcharts posted around the walls of their workspace and carried out the customisation in the centre of the room at workstation with a collection of Metavision manuals. A second workstation was used by IMDsoft staff who were present as trainers at the start of the one week initial customisation period.

The professional and systematic approach to end-user development was not accidental. Metavision training material has an explicit focus on project management, including recruiting suitable members of the development team, and establishing a systematic development process. This typically starts with the definition of "parameters" (patient data and measurements), followed by layout of a patient "flowsheet" that provides interactive access to various data entry forms as well as charts and reports. It is clear that the Metavision product takes as its starting point the importance of end-user software engineering (EUSE) concerns, somewhat in contrast to the Centricity training material, which at the time we saw it had a far more conventional focus on end-user programming (EUP) facilities.

From the perspective of a computer science researcher, the Metavision documentation was rather irritating, because basic description of language syntax and library functions are located in obscure parts of the documentation. The Centricity documentation offered far more conventional programming reference – perhaps a sign that the two audiences, end-user software engineering and end-user programmers should be clearly separated. In other EUP research in our group, we encourage the use of personas to distinguish between the two, characterising the approaches to programming that might be found among different professional groups [8].

4 EUD Application Structure

End-user development activities in Metavision are largely structured around the predefined dataflows and interaction model at the core of the product. It would not be possible for users to modify these system behaviours. Customisation of certain operations can be done using a scripting language that is invoked during particular operations, which include "triggers," relatively advanced messaging, action and automated notification functions. During the initial customization period, the team focused on creating parameters and forms rather than on triggers. However, they did use scripting facilities to write "formula" scripts that can calculate new parameters derived from directly captured data.

Formula scripts provided the main opportunity for us to observe conventional programming activity. IMDsoft training staff were able to assist with script syntax, but this threw attention onto the need for a shared domain understanding between technical and practical expertise. We observed a sustained debate over the interpretation of physical dimensions and data types, as clinical staff and trainers disagreed whether the built-in type "millimoles" represented a concentration or a quantity of potassium (ion balance in ICU patients was a key clinical concern of the ICU director).

A less contentious, although laborious, consequence of using formula scripts to implement local clinical concerns was the process of creating many formulas to calculate patient fluid balance – almost all drug administration and nutrition introduces fluids into the patient that must be taken into account alongside ion balance. As the

base product did not anticipate this particular cross-cutting concern, customisation activity had to include the creation of many small scripts throughout the system to account for fluid intake.

We also observed the consequences of conflict between the built-in operational model of the software, and the operational conventions of existing paper-based data processing. One of the primary displays in the Metavision software is a chart that can be customised to plot various parameters along a continuous timeline. The navigation of the chart and control of the timeline itself are sophisticated, and of course cannot be customised by end-users. However, this seemed to cause a major obstacle for the nursing staff. The local clinical concern with continuous monitoring of fluid balance meant that nurses were accustomed to noting that a patient had passed a quantity of urine, and attributing this retrospectively to a loss of fluid over the previous hour. However, the underlying model of the Metavision timeline seemed to assume that all observations related to events in the current, rather than previous time period.

The above three examples demonstrate a number of problems. The first and the third note the issue of having a sufficiently shared understanding between technicians and end-users to achieve a task. In the latter example particularly, neither the team nor the IMDsoft staff found it easy to identify this fundamental difference in the way they described the relationship between observations and intervals. Most likely, a common level of description would have required a shift to a more abstract conception of time [8]. The second and third examples indicate the importance of appropriate design even when customisation abilities are present. Although many problems can be worked around, as in the second example of writing many scripts to calculate fluid balance, others, such as the notion of time and flow, are more problematic. Research into requirements gathering for EUD systems that distinguishes between items that can and cannot be customised would be useful.

5 Studying Long-Term EUD Usage

In the second section we highlight the need to focus not just on the initial process of end-user customisation (as in cases two and three) but also on long term usage, enabling us to understand how customization becomes embedded in the social context. Lieberman et al [9] also stress the need for empirical studies of long-term EUD usage in their vision of EUD research of the next 15 years. This vein of research aims to answer questions ranging from when software is customised as opposed to when workarounds are found, to what role does cooperation play in the customisation process. We are now commencing a retrospective study of the ICU unit described above, investigating the day-to-day process of the customisation process of the Metavision system. This section describes how we intend this to be done.

5.1 Background

The majority of work that focuses on usage of EUD systems is carried out during the design, rather than evaluation phase. Rode et al. [10] for example, look at what features non-professional web-designers use in order to build an EUD system that addresses the needs of this particular group. Stevens et al. [11] focus more on the social

context of system use, describing problems with data control between two organizations and how that can be accounted for in an EUD system. In contrast, we concentrate on the long-term usage of the system. One expects a significant amount of customisation when a system is first bought and used. When, by whom, and for what purpose is the system customised after the initial burst? Noting Blackwell's attention investment theory of abstraction use [12], if users are given the opportunity to customise, when do they decide to adapt themselves rather than the system? How does the technology and the social context change the answer to this question?

5.2 Study Design

We have chosen two complementary approaches to address these questions: (1) a catalog of customised elements; (2) contextual interviews about specific changes made. The catalog will be used to explore patterns in what, when, and who does the customisation, while the contextual interviews are intended to investigate the social context in which customisation happens. The interviews will be broken into two parts, the first focusing on understanding the context for making changes and the second one concentrating on how the EUD capabilities are employed to achieve changes.

The catalog will be comprised of a database of all changes made in the past two years, documenting what change was made, when, and by whom (if possible). After categorizing the changes using grounded theory, it will be possible to look for patterns. We are particularly interested in whether there will be correlations between any of the following:

(1) the time elapsed since the initial customisation and the number of changes made
(2) time and what kind of changes are made
(3) time and who makes changes
(4) what changes are made by whom

We would also like to know if repeated changes are made to the same element. Not only will this data give us a general overview of customisation over time, but will provide fodder for the contextual interviews.

In the first part of the interview, the respondent will be asked about four changes in the catalog. Three will be randomly chosen to sample a range of possible user concerns and the fourth will be one of interest to the researchers, such as an element that has been changed several times. We have chosen to discuss concrete examples for two reasons. First, we want to understand the average case, rather than just extraordinary ones that are likely to be remembered. Second, it is usually easier for people to recall something specific (e.g. the change of the haemoglobin parameter) rather than something more general (what kinds of parameters have changed). The respondent will also be given an opportunity to discuss any significant changes that they remember in order to identify the most burdensome problems.

The questions used for both sections of the first part are listed below. They are open-ended and thus designed to elicit more information than is strictly implied in the question. Other questions will be used as necessary to develop themes that emerge.

1) Who made this change? (Does this person usually make changes?)
2) Why was this change made?

3) How was the decision made to make this change?
4) Were there any difficulties making this change?
5) Was the change tested after it was made?
6) How was the change communicated to the medical staff?
7) Was the accuracy of the change ever questioned?

The first question hopes to draw out comments about how people think of those making changes and perhaps how that affects their attitude towards or description of the changes. The second question will help us understand when changes are made and perhaps when they are not, and whether this varies over time. Questions 3 & 6 are oriented towards discovering the official and unofficial policies for making and distributing changes. Question 4 is an initial question to probe difficulties, either social or technical ones, which arise during customisation. Lastly, question 7 is an exploration of the social context of customisation.

The second part of the interview looks more specifically at interaction with the customisation interface and language. Again we employ the strategy of talking about concrete episodes, this time asking them to recall a specific incident. We also use a strategy of repetition, making the later questions refinements of the earlier. The questions are meant to address the social and technical issues raised when non-programmers use a programming language.

1) Is there any change that you would like to make but are unable? If so, can you describe how you would like to make this change?
2) In the past have there been problems that have been difficult to solve? What did you do?
3) Do you work on customisation with your colleagues?
4) How is this work similar and/or different from designing the paper charts?
5) Do you use metaphors or images or other aids when customizing?

6 Conclusion

We have presented a description of research work in progress, investigating the relationship between existing professional customisation practices, and software-based EUD practices, in a medical environment. It is clear that current commercial products already incorporate relatively sophisticated EUD facilities, and that these (unsurprisingly) are drawing attention to the importance of end-user software engineering, both in the formal interventions of software vendors, and the informal appropriation of technical capabilities within clinical teams. As such, we find that these experiences offer a valuable case study for comparison to experiences of EUD in educational or research contexts. Furthermore, they offer an opportunity for the long-term observation and analysis that is still in progress. This kind of long-term professional deployment is unusual in research contexts, and includes longitudinal study of individuals that is unusual in educational contexts (where studies of individual students usually last for a year at most). We believe that this research context will offer a valuable opportunity for further investigation of EUD in practice.

References

1. Schon, D.A.: The Reflective Practitioner: How professionals think in action. Basic Books (1983)
2. Martin, D., Rouncefield, M., O'Neill, J., Hartswood, M., Randall, D.: Timing in the Art of Integration: That's How the Bastille Got Stormed. In: Proc. of Group 2005 (2005)
3. GE Healthcare: Centricity EMR product description (August 29, 2008), http://www.gehealthcare.com
4. IMDsoft MetaVision MVICU product description (August 29, 2008), www.imd-soft.com
5. Bowker, G.C., Star, S.L.: Sorting Things Out: Classification and Its Consequences. MIT Press, Cambridge (1999)
6. Burnett, M., Myers, B., Rosson, M.B., Wiedenbeck, S.: The next step: from end-user programming to end-user software engineering. In: CHI 2006 (extended abstracts)
7. Orrick, E.: Position Paper for the CHI 2006 Workshop on End-User Software Engineering (2006), http://eusesconsortium.org/weuseii/docs/ErikaOrrick.pdf
8. Blackwell, A.F., Church, L.E., Green, T.R.G.: The abstract is an enemy. In: Proc. of PPIG (2008)
9. Lieberman, H., Paterno, F., Klann, M., Wulf, V.: End-User Development: An Emerging Paradigm. In: Lieberman, Paterno & Wulf (2006)
10. Rode, J., Rossen, M.B., Perez Quinones, M.A.: End-User Development of Web Applications. In: Lieberman, Paterno & Wulf (2006)
11. Stevens, G., Quaisser, G., Klann, M.: Breaking it up: An Industrial Case Study of Component-Based Tailorable Software Design. In: Lieberman, Paterno & Wulf (2006)
12. Blackwell, A.F.: First steps in programming: A rationale for Attention Investment models. In: Proc. of the IEEE Symposium on Human-Centric Computing Languages and Environments (2002)
13. Lieberman, H., Paterno, F., Wulf, V. (eds.): End-User Development. Springer, Heidelberg (2006)