Clinical Documentation as End-User Programming

Adam Rule[1], **Isaac H. Goldstein**[2], **Michael F. Chiang**[1,2], **Michelle R. Hribar**[1,2]
[1]Medical Informatics & Clinical Epidemiology, Oregon Health & Science University
[2]Casey Eye Institute, Oregon Health & Science University
{rulea, goldstis, chiangm, hribarm}@ohsu.edu

ABSTRACT

As healthcare providers have transitioned from paper to electronic health records they have gained access to increasingly sophisticated documentation aids such as custom note templates. However, little is known about how providers use these aids. To address this gap, we examine how 48 ophthalmologists and their staff create and use *content-importing phrases* — a customizable and composable form of note template — to document office visits across two years. In this case study, we find 1) content-importing phrases were used to document the vast majority of visits (95%), 2) most content imported by these phrases was structured data imported by data-links rather than boilerplate text, and 3) providers primarily used phrases they had created while staff largely used phrases created by other people. We conclude by discussing how framing clinical documentation as end-user programming can inform the design of electronic health records and other documentation systems mixing data and narrative text.

Author Keywords

end-user programming; electronic health record; text input

CCS Concepts

•**Human-centered computing** → **Interaction paradigms;** *Text input;*

INTRODUCTION

After most patient encounters, physicians write a note summarizing the visit. These notes help providers coordinate and communicate about care and are increasingly used to support billing, quality improvement, and research. While physicians have kept notes for millennia [19, 39], the content and structure of their notes have changed dramatically over the past few decades due to legal and technological changes. As physicians in the United States have migrated from paper to electronic health records (EHRs), regulators and insurers have simultaneously required they keep increasingly detailed notes [12, 31]. As a result, physicians in the US now spend up to half their time interacting with EHRs rather than patients [1, 54].

CHI '20, April 25–30, 2020, Honolulu, HI, USA.
2020 Association of Computing Machinery.
ACM ISBN 978-1-4503-6708-0/20/04 ...$15.00.
http://dx.doi.org/10.1145/3313831.3376205

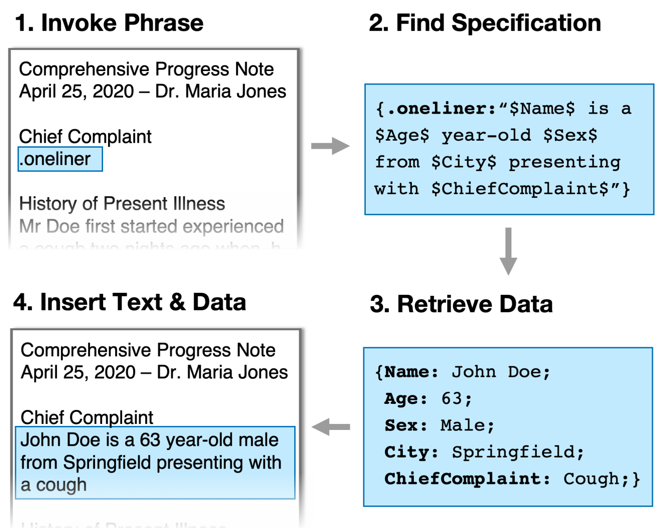

Figure 1. Physicians increasingly use *content-importing phrases*—a composable and customizable form of note template—to write clinical notes. 1) Typing a phrase's keyword into a draft note prompts the electronic health record to 2) look up the phrase's specification including boilerplate text and data-links (e.g., $Name$), 3) retrieve data from the patient record and 4) insert text and interpolated data into the note.

To cope with this growing documentation burden, providers have adopted diverse documentation practices. These include copying text from previous notes, using forms to simultaneously capture structured data and generate formulaic notes, and using templates to auto-populate notes with boilerplate text and previously captured patient data [59]. While many studies in the field of medical informatics have examined how frequently providers copy note text and the impact this practice has on data quality and patient safety [24, 53, 55], few studies have examined template or form use in similar detail [52, 57, 60]. Human-computer interaction (HCI) researchers have likewise examined how EHR adoption alters clinical workflows and spawns informal documentation practices, but paid less attention to how providers use electronic aids to produce the formal patient record [18, 45].

To narrow this gap, we present a case study of how 48 physicians and 200 staff in the ophthalmology department of a large academic medical center create and use *content-importing phrases* — a composable and customizable form of note template — to document 200,000 outpatient visits across two years. These phrases enable users to type keywords such as

.oneliner into draft notes to insert both boilerplate text and interpolated patient data (Figure 1). Due to their syntactical convention of starting with a period, content-importing phrases are commonly referred to as *dotphrases*.

In this case study, we examine how these phrases are created and used, common ways they are structured, and how their use varies from provider to provider and over time. Through this analysis, we observe documentation practices that routinely involve end-user programming. We argue that framing clinical documentation as end-user programming has implications not only for the design of EHRs but also documentation systems outside healthcare that mix narrative text and data.

This work has three primary contributions:

1. A case study of note template use at one academic ophthalmology department
2. Evidence that clinical documentation in this context routinely involves end-user programming
3. Implications of framing clinical documentation as end-user programming for the design of EHRs and other documentation systems mixing narrative text and data

BACKGROUND

Separating Narrative from Data

The earliest medical records were short narrative accounts preserved by ancient Greek and Egyptian physicians to instruct medical students [39]. Some of these accounts provide practical advice for treating common maladies; others document difficult or puzzling cases. As an example, one case in Hippocrates' *Epidemics* from the 5th Century BC opens "The wife of Dromeades having been delivered of a female child, and all other matters going on properly, on the second day after was seized with rigor and acute fever..." and ends a few sentences later with the blunt observation that "shortly afterwards spasms from the head began, and she immediately expired" [22].

Over the ensuing centuries, clinical documentation gained structure and content as medicine became more empirical and institutionalized. Simultaneously, notes began to be captured at the bedside to support patient care rather than just education after the fact. In the 19th century the traditional narrative case report fragmented into a set of standardized documents separating providers' subjective assessments from structured data such as lab results and medication lists [19]. This legacy is still apparent in modern records which separate structured data from physicians' textual notes which reference and interpret that data. Despite an increasing focus on objective data, the ability to form and work with clinical narratives (i.e., *narrative medicine*) is still acknowledged as a key clinical skill [10].

Impact of Regulation and EHRs

Over the last decade, clinical documentation has experienced another major transition as regulation, such as the Health Information Technology for Economic and Clinical Health (HITECH) Act in the Unites States, has incentivized a transition from paper to electronic health records [6]. Additional regulation has sought to increase the utility of EHR documentation by requiring clinical notes include specific information to support billing and auditing [4]. For example, the Evaluation and Management codes used by the Centers for Medicare and Medicaid Services stipulate that clinicians should document having reviewed ten different patient systems (e.g., eyes, ears, nose) to receive certain levels of reimbursement [36]. This requirement has led physicians to include "clinically meaningless" phrases such as "ten-point review of systems returned negative" or lists of default physical exam findings in their notes [31]. It is now common for clinical notes in the United States to exceed 4,000 characters in length (roughly the amount of text on this page), much of it tables and boilerplate text [15]. This is twice the length of notes produced before passage of the HITECH Act a decade ago and substantially longer than notes currently produced in other countries [15].

Researchers in HCI and medical informatics have examined how transitioning from paper to electronic health records has affected medical care in a number of ways [18, 45]. These studies have noted increases in documentation time [16, 40, 44], adoption of more comprehensive and structured data entry [21, 40], use of informal paper notes to workaround EHR constraints and coordinate care [11, 41, 65, 66], and numerous other unintended consequences [2]. Broadly, these findings reflect the tension between using medical records to coordinate care versus keeping a formal record of it [5, 14].

Content-Importing

Despite many studies examining documentation workflows, especially paper workarounds, little is known about how providers use electronic aids to produce the formal patient record. Rather than manually type every word of every note, providers now rely on a variety of "content-importing" technologies available in many EHRs to construct their notes [34, 59]. These include relatively simple tools such as those enabling selective or entire copying of prior notes (e.g., copy-paste, copy-forward) [24, 55], as well as more sophisticated structured data entry forms which generate formulaic notes (e.g., smart forms) [52], or templates that populate notes with both static text and previously captured patient data (e.g., templates, macros, data-imports) [57, 60]. Historically tools in this last category (which we collectively refer to as templates) were defined by system administrators and represented an entire note so only one template could be used at a time [42, 49, 57]. Now EHRs support inserting boilerplate text and patient data into notes in a variety of ways which differ in how templates are created, customized, and used.

In particular, many note templates are now implemented as *content-importing phrases* which providers invoke by typing keyphrases into their notes. These typically start with a period to differentiate them from normal note text. For example, typing the phrase *.oneliner* into a note and hitting enter might insert the text "John Doe is a 63 year-old male from Springfield presenting with a cough". Behind the scenes the EHR looks up the phrase's specification (e.g., "$NAME$ is a AGE year-old SEX from $CITY$ presenting with $CHIEFCOMPLAINT$") and uses the embedded data-links (e.g., $NAME$) to pull data from elsewhere in the patient record. So long as the note has not been signed, users might be able to manually refresh each field of imported data. These templates may also take parameters (e.g., number of most recent lab values to retrieve) or include syntax to embed drop-down lists into the note editor from which users can select one of multiple text options (e.g., the patient is <calm, anxious, combative>).

Several features differentiate the design and use of content-importing phrases from previous forms of note template. First, they are invoked by typing keywords into the note editor rather than selecting a template from a drop-down. Second, they are composable; more than one content-importing phrase can be used in a single note, enabling users to invoke different templates for different sections of their notes and at different stages of note editing. Finally, providers are often able to customize and create their own content-importing phrases in ways that were previously restricted to EHR administrators.

While a number of studies have examined how often providers copy-paste note text and the impact this practice has had on note quality and patient safety [24, 53, 55, 59], there have been few empirical studies describing the use of other electronic documentation aids such as templates [35, 57, 60], even as many papers and commentaries reference their use [12, 16, 24, 31]. Tellingly, recent studies that implemented standardized note templates for pediatric and internal medicine residents highlighted a lack of guidance for template design, even as there are best practices emerging around use of copy-paste [3, 27]. This lack of research led the American College of Physicians to recently call for researchers to *"study the authoring process and encourage the development of automated tools that enhance documentation quality without facilitating improper behaviors"* [31].

End-User Programming

This paper argues that use of content-importing phrases involves end-user programming and that this framing has implications for the design of EHRs and other data-rich documentation systems. End user programming is "programming to achieve the result of a program primarily for personal, rather [than] public use" [29]. That is, rather than producing generalizable code for wide use, end-user programmers write programs to help themselves or a select group of other people achieve a non-programming goal. This might include writing a script to apply the same post-processing steps to a large batch of photos, or creating a spreadsheet to compute grades for a course [29].

Historically, research on end-user programming focused on domains such as script-writing, scientific computing, and use of spreadsheets [8, 29]. More recently, studies have considered how end-users might manage Internet of Things (IoT) devices or automate tasks through trigger-action programming [13, 46, 56]. In their 2011 review of end-user software engineering, Ko et al. listed twenty classes of people who perform end-user programming [29]. While "healthcare workers" are included in this list, only one example is given, that of healthcare workers creating custom data-entry forms. While prior studies of other domains (e.g., scientific research) may provide some insight into how healthcare workers might leverage end-user programming, healthcare workers have different goals and face unique institutional and technological constraints which may shape their programming in unique ways, warranting separate study.

While focusing on applications outside healthcare, prior work on end-user programming has generated a number of overarching conceptual and theoretical frameworks to guide the study and design of end-user programming systems. For example, Ko et al. identify six learning barriers in end-user programming systems (i.e., Design, Selection, Coordination, Use, Information, Knowledge) which might hinder their use [30]. Researchers have also found it productive to consider how end-users might adopt software engineering practices such as testing and debugging to improve the robustness of their programs, a paradigm known as end-user software engineering [8, 29]. Finally, prior work has also noted the variety of programming paradigms which might support end user programming (e.g., visual programming, block programming, programming by example) and noted the need for task or domain-specific languages to support end-users' highly specialized work [29, 37, 38]. We return to these frameworks and their implications for design in the Discussion.

One widely studied form of end-user programming warrants further comment due to its similarity to the paradigm observed in this study, that of *mashups*. Mashups are a type of application, typically a web application, that "combines data, either through APIs or other sources, into a single integrated user experience" [64]. For example users might create an application that scrapes Craigslist for the location of apartment listings and uses the Google Maps API to plot them on an interactive map. While the form of end-user programming examined in this paper is more limited (i.e., string interpolation pulling data from a single API and supporting minimal interactivity), as with other mashup platforms the end goal is to pull data from disparate locations and present it in a useful combined form.

The emphasis of early mashup research on enabling novice programmers to generate complex applications through diverse programming paradigms may also provide some insight into how EHR templates might support more complex interactions with patient data [32, 62]. For example, one prior line of work explored how providers might use a block-based programming language to generate clinical notes with different content and formatting based on logical operators, though providers only interacted with a prototype of this system in design sessions [9]. We return to this point in the Discussion.

Building on this long history of prior work, this case study advances our understanding of clinical documentation by providing empirical evidence about how one form of note template is being customized and used by providers and their staff. It also demonstrates how clinical documentation can routinely involve end-user programming. Combined with prior work on novel note-writing interfaces [9, 61], these observations suggest it may be productive to frame clinical documentation as end-user programming and that providers may be willing to adopt other forms of end-user programming to support both formal and informal documentation. Framing clinical documentation as end-user programming may also have implications for documentation systems outside healthcare that increasingly mix narrative and data.

METHODS

Setting

This study was conducted in the ophthalmology department of a large academic medical center on the United States' west coast and approved by that center's Institutional Review Board. The department includes over 50 faculty providers and 200 staff who conduct more than 130,000 office visits every year. Providers specialize in one of a dozen different areas of ophthalmology (e.g., Comprehensive, Cornea, Glaucoma, Pediatrics, Retina) many of which require completing a 1-2 year fellowship in addition to medical school and residency. Staff at the study site include *technicians* who typically have an ophthalmology-specific associate degree or certificate and assist with rooming patients and performing routine ophthalmic exams (e.g., visual acuity and visual field tests). In high volume clinics staff may also include *scribes* who assist with documentation during the encounter and are typically either retrained technicians, pre-medical university students seeking medical experience, or hired from a scribe company [7].

Providers at our study site typically see patients during one or two four-hour "clinics" each day with the number of patients seen per clinic varying by specialty. Providers write progress notes summarizing each visit using an ophthalmology-specific module of a large commercial EHR (Kaleidoscope; Epic Systems, Verona WI) and often collaborate with their staff to write these notes. A typical documentation workflow might involve a technician documenting exam findings in structured fields of the EHR while performing each exam and then starting a draft progress note for the visit using a template that imports this newly captured structured data. This might be followed by a scribe listening to the patient-provider interaction as the physician examines the patient, adding text to the draft note describing the provider's assessment of the patient and care plan. Finally the provider may review, edit, and sign the scribe's draft note sometime after the visit. While this is a canonical workflow, not all physicians at our study site employ scribes or have technicians assist with writing the note. Moreover, some providers practice other forms of documentation such as "problem-based charting" which integrates note writing with managing the patient's problem list in the EHR.

We chose to examine use of content-importing phrases in this context as ophthalmology is a high-volume specialty where providers routinely write 20-40 notes per day, potentially leading to heavy use of content-importing. The distribution of documentation work between staff and providers also enables observation of collaborative documentation practices. Moreover providers at the study site have been using content-importing phrases since they first adopted an EHR in 2006, enabling observation of mature electronic documentation practices.

Data Collection

We collected data on every time a content-importing phrase was used in the department between January 1, 2017 and December 31, 2018. This included the time each phrase was invoked, the patient visit it was used to document, the context in which it was invoked (e.g., charting, letter writing), who invoked it, the phrase's keyword, the phrase's specification of text and data to import, and who originally created the phrase. We restricted our analysis to phrases that were 1) invoked by the 48 providers, or their staff, who saw patients across the entire two year study period, and 2) were invoked while charting, that is while writing a clinical note, rather than writing a letter to another provider or preparing patient instructions. We additionally collected data on each progress note written during the study period including the note's text, the note's length in words, and the percent of the note which had been imported by a template, copy-pasted, or manually entered.

Phrase Counts and Categories

We counted the number of times each unique phrase was invoked. Using the phrase's keyword and specification, one author manually categorized each phrase as either inserting 1) a full-note, 2) a signature or attestation, or 3) some other block of helpful text. To assess inter-rater reliability, a second author independently coded the 100 most frequently used phrases, achieving near-perfect inter-rater reliability (0.96 Cohen's Kappa). Figure 2 shows examples of each phrase type.

.VA – Helper Phrase

visual acuity

.TECHATTEST – Signature/Attestation Phrase

I, $USERNAME$, <performed, reviewed, revised> the above history, medications, allergies, as well as performed elements noted in the Base Ophthalmology Exam.

.SOAPPROGRESS – Full Note Template Phrase

COMPREHENSIVE OPHTHALMOLOGY PROGRESS NOTE

Assessment and Plan:
Exam Date: $DATE$
Patient:$NAME$ ($RECORDNUMBER$)

Impression:
***
$SPECIALISTCOMMENTS$

Plan:
***

Physician: $PROVIDERNAME$, MD $DATE$

HPI: $NAME$ ($RECORDNUMBER$), AGE year old SEX
$OCCUPATION$ from $CITY$:
$REASONFORVISIT$

***

Tobacco use: $TOBACCOHISTORY$
Primary Care Provider: $PRIMARYCAREPROVIDER$

Figure 2. Examples of the three types of content-importing phrases coded in this study. Phrase definitions include embedded tools such as text drop-downs (represented by the <option 1, option2, ... > syntax), data-links (represented by the $LINKNAME$ syntax), and placeholders for manual text entry (e.g., *).**

Phrase Length and Embedded Tools

Content-importing phrases can include not only static text and links to import data from the patient record but also drop-downs from which to select one of a number of static phrases

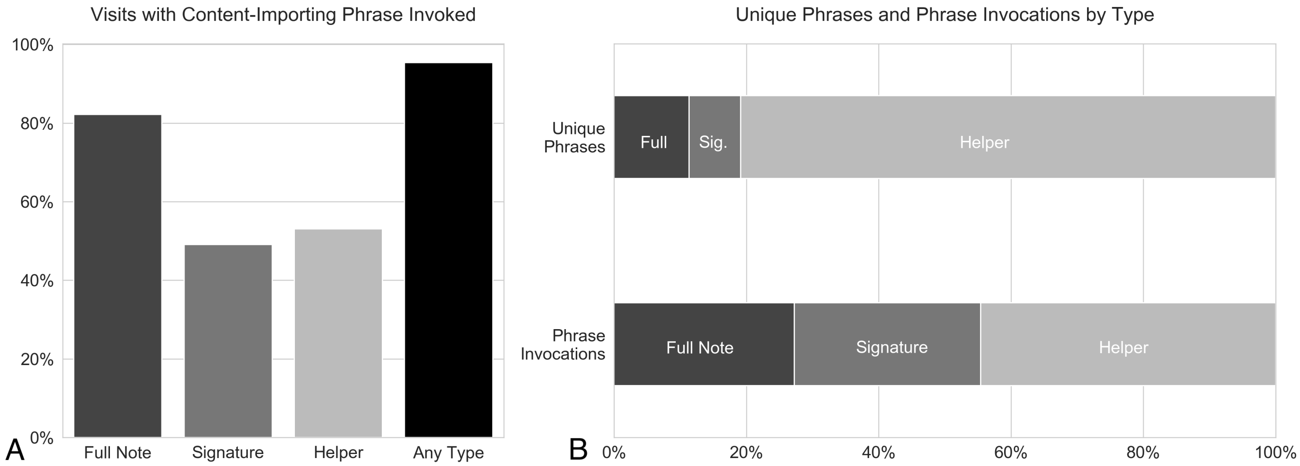

Figure 3. Frequency of phrase use by phrase type. A) Percent of visits with content-importing phrase used for documentation by phrase type (i.e., full-note templates, signature lines, short helper phrases) B) Distribution of unique phrases and phrase invocations by phrase type.

(e.g., patient was <alert, clam, agitated>) and manual entry placeholders which users can tab between when editing inserted text (e.g., ***). We counted the number of data-links, text drop-downs, and manual-entry placeholders embedded in each phrase as well as the number of static words each phrase imported (i.e., those not populated via a data-link or text drop-down). We also looked for evidence of users passing parameters to phrases at the time of invoking them (e.g., invoking *.lastbp(3)* to insert the patient's last 3 blood pressure readings into the note) or having similarly parameterized data-links embedded in the phrases they used.

We estimated the percent of text in each note that had been imported via data-links. For each note created with the help of at least one content-importing phrase, we subtracted the total number of static text characters in content-importing phrases used during that visit from the total number of characters imported into that note using templates as tracked by the EHR. We divided the resulting value by the total number of characters in the note to estimate the percent of note text imported by data-links. To inspect text imported by data-links in more detail, for each of the 50 most frequently used data-links we randomly selected 100 notes generated using a content-importing phrase containing that data-link and manually counted the number of words imported by the link. We also compared the data-links embedded in the 50 most frequently used full-note templates.

Patterns of Phrase Creation and Use

Lastly, we examined patterns of phrase use. We counted the number of phrases invoked during each patient visit (both total phrases and by phrase type) as well as the number of distinct people who used each phrase over the study period. We compared the number of times phrases were invoked by a provider versus a staff member, and how often each provider or staff member invoked phrases they had created versus phrases created by other people. Finally, we examined whether providers switched which full-note templates their team used over time and, if a switch had occured, compared the average length of notes written using each template.

RESULTS

Phrase Counts and Categories

We identified 5,144 unique content-importing phrases invoked a total of 647,524 times across the 200,695 office visits included in this study. The vast majority of visits during our study period were documented using at least one content-importing phrase (95.4%), with full-note phrases invoked during 82.1% of visits (Figure 3A). The 4.6% of progress notes written without using a content-importing phrase were on average shorter (469 vs. 611 words), contained substantially more copied text (29% vs 13% of note text), and were primarily written by just 5 of the 48 study providers.

Full-note and signature phrases accounted for over half of the 647,524 phrase invocations during the study period (27.3% and 28.2% respectively), but were a small percentage of the 5,144 unique content-importing phrases (11.3% and 7.8%) (Figure 3B). Eight of the ten most frequently used phrases were full-note templates while the most frequently used phrase was the technician attestation shown in Figure 2.

Phrase Length and Embedded Tools

The number of words inserted by each phrase and number of tools embedded in each phrase varied by phrase type (Table 1). Full-note phrases imported the most static text (i.e., text not imported by embedded data-links or text drop-downs), with a median of 81 words (67-105 Interquartile range) followed by signature phrases with 22 words (17-31 IQ) and helper phrases with 4 words (2-13 IQ). Full note template phrases included a median of 18 data-links (13-21 IQ) and 4 manual-entry placeholders (2-6 IQ) whereas signature and helper phrases rarely included data-links or placeholders. In all, 44% of unique phrases included a data-link and 95% of all visits involved using one of these phrases with an embedded data link. Due to limitations in how our EHR tracks phrase use, we were unable to observe when phrases were invoked with a parameter (e.g., *.lastbp(3)*). However, we were able to observe that 1% of phrases used during the study period employed paramaterized data-links such as $LASTLABRESULT(a1c)$.

	Full Note		Signature		Helper	
	median	*interquartile*	*median*	*interquartile*	*median*	*interquartile*
Static Words	81	67-105	22	17-31	4	2-13
Data-Links	18	13-21	0	0-1	0	0-0
Placeholders	4	2-6	0	0-0	0	0-0
Text Drop-downs	0	0-3	0	0-0	0	0-0

Table 1. Characteristics of observed phrases by phrase type (i.e., full-note templates, signature lines, or short helper phrases) including words of boilerplate text, links importing patient data (i.e., data-links), placeholders for manual text entry, and drop-downs for selecting one of many text options

The 50 most common data-links accounted for 92.4% of all data-links embedded in content-importing phrases. The amount of text imported by data-links varied significantly both within and between links. For example, whereas the links for a patient's AGE and SEX only ever imported one word, one link for medication lists imported 100 $\pm$ 177 words, sometimes importing no text and other times importing more than 350 words depending on the patient's prescribed medications. Moreover, multiple data-links performed the same function, importing exactly the same data only with slightly different headers (e.g., "Current Medications" vs. "Medications" followed by the same table of medications). Three of the 50 most used links specified nearly identical ways to import the patient's reason for visiting and three imported similarly isomorphic medication lists. While an individual template would only use one of these isomorphic links, different templates used different links to import the same data. We estimated that text inserted by data-links accounted for nearly half of all text in notes created with the help of at least one phrase (43.4%, [43.2, 43.7] 95% CI), and the majority of all text imported by content-importing phrases (62.5%, [62.5, 62.6] 95% CI).

Patterns of Phrase Creation and Use

Providers and their staff invoked a median of 3 content-importing phrases each visit (2-4 Interquartile range), representing a median of one full-note phrase (1-1 IQ), one signature/attestation phrase (0-2 IQ), and one phrase that imported some other helpful snippet of text (0-2 IQ). However, the number and type of phrases used per visit varied by provider. Whereas some providers and their staff tended to use just one phrase per visit, others used up to an average of fourteen.

Most phrases were invoked by providers (58.1%) as opposed to their staff, though this varied by provider and phrase type. Whereas some providers invoked nearly all phrases used to document their patients' visits (99.7% provider invoked), other providers had their staff invoke nearly all phrases (0.8% provider invoked). Moreover, while staff invoked the majority of full-note phrases (40.0% provider invoked) providers invoked most helper phrases (74.6% provider invoked).

Providers invoked an average of 108 unique phrases across the two years of the study whereas staff members invoked an average just 23 unique phrases. The vast majority of both were short helper phrases such as ones that inserted common strings of text like "right eye" or "hours per day". In an extreme case, one provider employed 508 unique phrases over the course of the two year study. Most phrases that providers invoked were ones that they had created (71.8% user generated), whereas most phrases invoked by staff had been created by other people (40.9% user generated). Still, a full 46 of the 48 study physicians (96%) and 148 of the 197 study staff (75%) created and used at least one custom phrase.

Individual providers and their staff tended to invoke only two full-note template phrases across all visits regardless of the patient's chief complaint or diagnosis: one for new patient visits and one for return patient visits. Focusing on phrases that generated full-note templates for return patient visits (i.e., the majority of study visits) revealed that while some providers shared templates, many providers (16 of 48) had their own unique full-note templates that only they and their staff used. Comparing the 50 most frequently used return visit note templates (which accounted for over 99% of all return template invocations) we found that on average two randomly sampled templates shared 45% of their data-links (e.g., 8 of 18 links), with the links for the patient's name, sex, age, and reason for visit most likely to be shared between templates.

Examining phrase use over time, we identified four providers who switched their full-note template for return patient visits during the study period. After switching, each provider dramatically reduced their average note length by around 100-200 words, or 27-42% of the note's length. While in some cases the number of static words imported by the template actually increased between the old and new template, in every case the number of data-links went down. For example, two providers switched from templates that included data-links for the patient's past medical history, medication list, surgical history, and family history (among others) to templates that did not.

DISCUSSION

This case study of clinical documentation reveals widespread use of custom note templates invoked collaboratively to write clinical notes. Moreover, these results reveal widespread use of end-user programming in the creation and use of these templates. Framing clinical documentation as end-user programming suggests implications not only for the design of EHRs but also systems mixing data and narrative text outside healthcare, especially when viewed through the lens of frameworks for learning end-user programming [30], supporting end-user software engineering [8, 29], and leveraging diverse end-user programming paradigms [37, 38].

Use of Content-Importing Phrases

Content-Importing as Abstraction Work

Providers and staff at our study site used content-importing phrases to generate documentation for the vast majority of office visits (95.4%). This finding reinforces existing evidence that *clinical documentation increasingly involves use of electronic aids* rather than manual text entry. For example, one prior study of inpatient notes written in 2016 at a large academic medical center found 46% of note text was copied from prior notes, 36% was imported using tools such as templates, and just 18% was manually typed [58].

Routine use of templates likely reflects providers' desire to quickly construct notes that meet government and organizational standards. However the widespread customization of templates observed in this study may further reflect use of templates to *create working records that perform abstraction work*, enabling providers to quickly synthesize information from disparate locations across the record [17, 41]. Heath and Luff note that EHRs tend to fragment patient information, separating it by type (e.g., lab results, medication, diagnosis) rather than presenting it in coherent narratives organized around problems [21]. As a result, providers spend substantial time reconstructing a patient's story when using EHRs. Park et al. note how ED providers annotate paper notes to create succinct abstracts of the patient's situation, and that these abstractions vary by user role and department [41]. The frequent customization of full-note templates observed in this study may similarly reflect providers using content-importing phrases to create digital abstracts of the patient's record to aid synthesis. In one prior study, providers at the study site noted that they often include exam data in their progress notes not because it was needed for billing, but because this practice made it easier to find and review before the visit [25]. Viewing template use as abstraction work also aligns with prior work which found documentation is more accurately described as synthesis of existing data than composing from scratch [35].

Casting the creation of electronic documents (not just paper documents) as abstraction work suggests healthcare providers may benefit from better tools to search for and synthesize information from across the patient record. In addition to applications that provide rich but standardized visualizations of longitudinal patient records [23, 43], this might also involve developing tools that enable providers to interactively construct custom summaries from scratch. Content-importing phrases are a step in this direction, but might be designed and governed to support more granular search and synthesis rather than importing large batches of patient information at once, as this may hinder synthesis and make resulting notes difficult to review. Were documentation requirements less prescriptive, we might have observed less use of full-note templates and greater use of phrases importing limited sets of data relevant to the patient's diagnosis (e.g., a glaucoma workup vs. a cataract workup). This heavy reliance on full-note templates may also be a product of studying specialty rather than primary or emergency care where diagnoses are more varied.

Mixing Data and Narrative

Our results also revealed that much text imported by content-importing phrases was free-text representing structured data rather than static boilerplate text. Content-importing phrases coded as full-note templates in this study had a median of 18 data-links embedded within them which imported information such as names, dates, problems, and medication lists. Across all phrases, these data-links generated the majority of text imported by content-importing phrases (62.5%) and nearly half of all text (43.4%) in notes constructed using them.

These findings reflect both a desire to *mix data and narrative* in clinical documentation and the challenge of doing so. Prior work has noted the varied strengths of structured and unstructured data in clinical documentation such as supporting visualization and real-time decision support versus aiding comprehension and recall [47, 48]. Other work has developed systems to more closely integrate structured and unstructured data through documents based on XML and other markup languages [26, 50, 61]. EHRs might benefit from interactions and data-structures that enable providers to more fluidly incorporate structured data into their notes, such as by hyperlinking to or translucing information rather than copying it.

These findings also reflect both the power of data-links and their potential for misuse. On the one hand, data-links can be woven with static text to quickly produce detailed yet concise summaries (e.g., the *.oneliner* phrase in Figure 1). But data-links can just as easily bloat notes with uninformative or unnecessary text. Many note templates observed in this study included links for long medication and surgery lists which may not have been relevant to the patient's visit, or needed for billing. Indeed some providers switched to templates without these links during the study period. The amount of text imported by these links was also highly variable based on the availability of underlying structured data, sometimes importing no text, other times importing hundreds of words.

Despite their potential to quickly bloat notes, data-links provide a unique opportunity to *shape documentation practices at scale*. A single data-link may be embedded in hundreds of templates, enabling a small tweak in the way one link retrieves information to affect the documentation behavior of hundreds of providers. For example, a medication data-link could be revised to return only current medications or be context-sensitive and return only a list of medications relevant to the patients' chief complaint or provider's specialty with a link to the full list. How data-links might be improved is open to debate. And while it is likely easier to change how a single data-link returns text than ask dozens or hundreds of providers to manually change their templates, doing so without provider input might undercut the social construction of documentation practices needed to ensure they align with clinical workflows.

Customizing Documents and Workflows

Provider and staff use of content-importing phrases was fragmented and varied, revealing widespread customization of both phrases and documentation workflows. Whereas some providers used a single content-importing phrase at a typical visit, others used up to fourteen. Additionally, while in some groups the providers invoked nearly all the content-importing phrases, in other groups staff such as technicians and scribes invoked most of them while setting up notes for physicians to edit. Most strikingly, many providers had custom phrases producing nearly complete notes which only they and their staff used. While there was some overlap in the structured data (e.g., data-links) in templates used by different providers, a significant proportion of links were unique.

In creating custom content-importing phrases, *providers shape their notes and the process by which they and their staff produce them*. Typically, this process involved a staff member initiating a note using a full-note template, a provider filling in details of their assessment and plan using short helper phrases, and both providers and staff signing off on their respective contributions. This distribution reflects creating different phrases for different parts of the documentation workflow. However as noted above, this workflow was by no means consistent across providers. Some providers used scribes while others did not. Some practiced problem-based charting while others followed a more traditional note-centric workflow. Regardless of approach, it is striking to note that all providers used content-importing phrases at some point of most visits.

Overall, 72% of the phrases providers used were ones they had created, revealing a high level of customization as opposed to prior studies where template customization was more limited or not even possible by providers [57]. However, most of the phrases used by staff were created by other people. These findings may reflect the control individual providers have over how their group documents, or disagreements between providers about what information is vital to include in notes. It may also reflect a lack of tools supporting sharing and reuse of phrases created by others. For example, at our institution providers must search for phrases created by other people by first entering the name of the user whose phrases they wish to browse, rather than searching for phrases across users.

Clinical Documentation as End-User Programming

Together, these findings reveal how current EHRs enable physicians to program their notes by creating and invoking functions that perform string interpolation. Rather than manually type recurring note text and repeatedly search for and copy patient information, providers and their staff program notes by manipulating a small set of phrases which represent pre-determined blocks of text. These blocks may in turn have syntax embedded within them which invoke other content-importing technologies (e.g., data-links, text drop-downs) to fill out the note. In the case of text drop-downs, these embedded tools even modify the note editing interface by inserting UI elements for interactive specification of note content. While providers have access to phrases created by EHR vendors or their institution's information technology staff, many also have the ability to create their own. Viewing this process of clinical documentation through the lens of end-user programming suggests several implications for the design, governance, and use of clinical information systems.

Training and Software Engineering

At our institution, a significant portion of providers' EHR training is already devoted to creation of content-importing phrases. However, just as programmers receive instruction not only in the mechanics of programming but also in programming best practices, providers may need more guidance on how to structure and use phrases effectively, and how to overcome barriers to phrase use. For example, Ko et al. note six *barriers to learning* how to use end-user programming systems (i.e., Design, Selection, Coordination, Use, Understanding, Information) which apply to phrase use as well [30]. For example, when creating phrases providers may struggle to select which data-links to use or lack understanding of how those links operate when data are missing. Training could have additional benefits as EHR training and personalization have been associated with higher EHR satisfaction [33].

EHR trainers and designers might also consider how to support and encourage *end-user software engineering* through practices such as debugging and testing to make templates more robust [8, 29]. Testing tools might help providers identify which aspects of documentation regulation are fulfilled by free-text and data in their notes, either during phrase creation or phrase use. EHRs might also provide better support for when data-links don't retrieve information, helping users determine if the data simply do not exist for that patient, are poorly formatted, or stored in another part of the record.

Phrase Design and Governance

Building on the framing of clinical documentation as end-user programming, and more specifically end-user software engineering, EHR designers and system administrators may benefit from *treating both content-importing phrases and data-links like Application Programming Interfaces (APIs)*. As with other APIs, content-importing phrases and data-links need to be designed to have short, descriptive, and memorable syntax and users need to have access to clear documentation if they are to use them effectively. Currently many EHRs lack functionality to search for existing phrases or data-links or documentation describing what each does. This may lead to some users duplicating existing phrases or not making full use of their features, such as the ability to pass parameters to certain phrases or create drop-downs for specifying common findings.

While some content-importing phrases, data-links, and text drop-downs come standard with EHRs, many are specified by organizations during configuration or by providers during the course of care. While enabling providers to create their own content-importing phrases supports autonomy and localization [41], it also complicates maintenance, leading to fragmented and varied use as observed in this study. This fragmentation was not restricted to providers. Even the data-link definitions, which at our site only information technology staff can create, were redundant with several commonly-used links performing the same function.

While this duplication might be addressed with better search and sharing tools it also *raises the question of who should be responsible for creating and maintaining content-importing technologies and what kind of standardization might be beneficial*. With many providers practicing at multiple locations, even standardizing phrases at the clinic level may leave some providers having to recreate templates in other offices. Yet standardizing across entire departments might stifle localization. Which of these issues might be addressed through well-established software engineering practices such as code review, branching, and version control and which require domain-specific solutions remains an open question.

Exploring Other Programming Paradigms

Recognizing that clinical documentation increasingly involves end-user programming should also encourage EHR designers to think more creatively about how notes might be constructed in the future. Some interaction paradigms from other development environments such as tab completion, previews, inline documentation, and linting (e.g., highlighting when information to be imported is already in the note) might support more robust, granular, and fluid documentation that is simultaneously easier to write and easier to read. Moreover, designers might consider leveraging *other end-user programming paradigms* described in the literature such as visual programming, block-based programming [9], or programming by example, rather than relying on the current paradigm of using a domain-specific language in a text editor to construct notes [37].

As explored in prior work, clinical notes might also mix narrative text, rich information displays, and textual commands to both retrieve data and perform actions (such as place orders), similar to how computational notebooks support data science workflows [26, 50, 61]. Mixing data, narrative, and action in this way may better support the data manipulation canonical of clinical workflows, particularly if they are designed to support informal documentation practices used for sense-making and care coordination as well as construction of the formal record [20, 66]. However, technological changes would need to be supported by regulatory ones that enable providers to create data-rich informal digital documentation that does not form part of the legal record.

Recent regulatory efforts to reduce information blocking (e.g., siloing of EHR data) and support for the FHIR open standard for transfer of EHR data provide an unique opportunity for researchers in HCI to innovate on the design and operation of EHRs through easier creation of third-party apps that interact with EHR data. These reforms might also enable providers to generate their own documentation aids in more sophisticated forms of end-user programming as observed in prior work on web mashups.

Beyond Electronic Health Records

This research may also have implications for the design of other documentation systems that mix narrative and data, such as computational notebooks [28, 51]. For example, the paradigm of inserting blocks of text or code via keywords with placeholders for manual editing might assist with the documentation or performance of data-intensive analyses. Content importing phrases might save developer time by enabling them to quickly write the code for their standard set of library imports, or to include a comment documenting the version of every dependency used in their analysis. It might also encourage documentation and coding best practices by providing analysis schemas for common analytical steps [63]. Moreover it might lower the barrier to less experienced programmers generating interactive documentation in these environments.

Limitations and Future Work

This research has a number of limitations which future work could address. First, as a case study it examines documentation practices at a single academic ophthalmology department. Future work could complement this research by exploring content-importing in other disciplines and institutions, particularly those involving inpatient primary care. While the prevalence, frequency, and distribution of phrase use across team members is like to vary between institutions and specialties, we expect phrase use is still widespread in other medical contexts. This is based on evidence of a moderately high proportion of imported text in internal medicine notes at one large academic hospital (36%) [58] and studies reporting the implementation of standard note templates across multiple institutions in both internal medicine and pediatrics [3, 27], suggesting widespread use of custom templates in those disciplines. However, phrases in these contexts may tend to be organized around diagnoses rather than visit type.

Second, there are a number of documentation aids such as checkbox-driven note writers which this research does not investigate, but which may also enable providers to programmatically generate chunks of note text such as formatted exam findings. Third, this work has not investigated qualitative aspects of content-importing. Future work could explore what motivates providers to import content, how providers and staff collaborate to construct notes using these tools, and users' assessment of content-importing's strengths and weaknesses. Finally, future work could further explore the end-user programming paradigm in healthcare by developing new tools or examining other tasks (such as chart review) where providers or staff may already perform end user-programming in EHRs.

CONCLUSION

This case study advances our understanding of how clinical documentation is performed in EHRs by revealing widespread use of content-importing phrases — a customizable and composable form of note template — to generate clinical notes at our study site. We argue that creation and use of these phrases reflects the use of electronic documents to perform abstraction work (not just paper documents), a desire to mix narrative and data in clinical notes, and the simultaneous crafting of both clinical documents and clinical workflows.

By creating and using these templates which specify custom string interpolation functions, providers and their staff leverage the computing power of EHRs to generate detailed documents mixing text and data. While this lightweight form of end-user programming arises from a particular context of use, it has broader implications for the design of EHRs and other data-rich documentation systems when viewed through the lens of established end-user programming frameworks such as those describing learning barriers [30], programming paradigms [37, 38], and the need to support software engineering [8, 29].

Future work could explore how EHRs might support end-user programming in other aspects of clinical workflow, especially the generation of informal working documentation, and how other programming paradigms such as visual programming or the design of richer domain-specific languages might better support the narrative and data-driven work of clinical care.

ACKNOWLEDGMENTS

This research was supported by grants R00LM12238, P30EY10572, and T15LM007088 from the National Institutes of Health (Bethesda, MD), and by unrestricted departmental funding from Research to Prevent Blindness (New York, NY).

REFERENCES

[1] Brian G. Arndt, John W. Beasley, Michelle D. Watkinson, Jonathan L. Temte, Wen-Jan Tuan, Christine A. Sinsky, and Valerie J. Gilchrist. 2017. Tethered to the EHR: Primary Care Physician Workload Assessment Using EHR Event Log Data and Time-Motion Observations. *The Annals of Family Medicine* 15, 5 (2017), 419–426. DOI: http://dx.doi.org/10.1370/afm.2121

[2] Joan S. Ash, Marc Berg, and Enrico Coiera. 2004. Some unintended consequences of information technology in health care: the nature of patient care information system-related errors. *Journal of the American Medical Informatics Association: JAMIA* 11, 2 (2004), 104–112. DOI:http://dx.doi.org/10.1197/jamia.M1471

[3] Megan Aylor, Emily M. Campbell, Christiane Winter, and Carrie A. Phillipi. 2017. Resident Notes in an Electronic Health Record. *Clinical Pediatrics* 56, 3 (2017), 257–262. DOI: http://dx.doi.org/10.1177/0009922816658651

[4] Robert A. Berenson, Peter Basch, and Amanda Sussex. 2011. Revisiting E&M visit guidelines–a missing piece of payment reform. *The New England Journal of Medicine* 364, 20 (2011), 1892–1895. DOI: http://dx.doi.org/10.1056/NEJMp1102099

[5] Marc Berg. 1999. Accumulating and Coordinating: Occasions for Information Technologies in Medical Work. *Computer Supported Cooperative Work (CSCW)* 8, 4 (1999), 373–401. DOI: http://dx.doi.org/10.1023/A:1008757115404

[6] David Blumenthal. 2010. Launching HITECH. *The New England Journal of Medicine* 362, 5 (2010), 382–385. DOI:http://dx.doi.org/10.1056/NEJMp0912825

[7] Claus Bossen, Yunan Chen, and Kathleen H. Pine. 2019. The emergence of new data work occupations in healthcare: The case of medical scribes. *International Journal of Medical Informatics* 123 (2019), 76–83. DOI: http://dx.doi.org/10.1016/j.ijmedinf.2019.01.001

[8] Margaret Burnett, Curtis Cook, and Gregg Rothermel. 2004. End-user software engineering. *Commun. ACM* 47, 9 (2004), 53–58. DOI: http://dx.doi.org/10.1145/1015864.1015889

[9] F. Cabitza, I. Gesso, and C. Simone. 2012. Providing end-users with a visual editor to make their electronic documents active. In *2012 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, Innsbruck, 171–174. DOI: http://dx.doi.org/10.1109/VLHCC.2012.6344509

[10] R. Charon. 2001. The patient-physician relationship. Narrative medicine: a model for empathy, reflection, profession, and trust. *JAMA* 286, 15 (Oct. 2001), 1897–1902. DOI: http://dx.doi.org/10.1001/jama.286.15.1897

[11] Yunan Chen. 2010. Documenting transitional information in EMR. In *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*. ACM Press, Atlanta, Georgia, USA, 1787–1796. DOI: http://dx.doi.org/10.1145/1753326.1753594

[12] Caitlin M. Cusack, George Hripcsak, Meryl Bloomrosen, S. Trent Rosenbloom, Charlotte A. Weaver, Adam Wright, David K. Vawdrey, Jim Walker, and Lena Mamykina. 2013. The future state of clinical data capture and documentation: a report from AMIA's 2011 Policy Meeting. *Journal of the American Medical Informatics Association: JAMIA* 20, 1 (2013), 134–140. DOI:http://dx.doi.org/10.1136/amiajnl-2012-001093

[13] Luigi De Russis and Fulvio Corno. 2015. HomeRules: A Tangible End-User Programming Interface for Smart Homes. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '15*. ACM Press, Seoul, Republic of Korea, 2109–2114. DOI: http://dx.doi.org/10.1145/2702613.2732795

[14] Paul Dourish. 2001. Process descriptions as organisational accounting devices: the dual use of workflow technologies. In *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work - GROUP '01*. ACM Press, Boulder, Colorado, USA, 52–60. DOI: http://dx.doi.org/10.1145/500286.500297

[15] N. Lance Downing, David W. Bates, and Christopher A. Longhurst. 2018. Physician Burnout in the Electronic Health Record Era: Are We Ignoring the Real Cause? *Annals of Internal Medicine* 169, 1 (2018), 50–51. DOI: http://dx.doi.org/10.7326/M18-0139

[16] P. J. Embi. 2004. Impacts of Computerized Physician Documentation in a Teaching Hospital: Perceptions of Faculty and Resident Physicians. *Journal of the American Medical Informatics Association* 11, 4 (2004), 300–309. DOI:http://dx.doi.org/10.1197/jamia.M1525

[17] Geraldine Fitzpatrick. 2004. Integrated care and the working record. *Health Informatics Journal* 10, 4 (2004), 291–302. DOI: http://dx.doi.org/10.1177/1460458204048507

[18] Geraldine Fitzpatrick and Gunnar Ellingsen. 2013. A Review of 25 Years of CSCW Research in Healthcare: Contributions, Challenges and Future Agendas. *Computer Supported Cooperative Work (CSCW)* 22, 4-6 (2013), 609–665. DOI: http://dx.doi.org/10.1007/s10606-012-9168-0

[19] Richard F. Gillum. 2013. From papyrus to the electronic tablet: a brief history of the clinical medical record with lessons for the digital age. *The American Journal of Medicine* 126, 10 (2013), 853–857. DOI: http://dx.doi.org/10.1016/j.amjmed.2013.03.024

[20] Gillian Hardstone, Mark Hartswood, Rob Procter, Roger Slack, Alex Voss, and Gwyneth Rees. 2004. Supporting informality: team working and integrated care records. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work - CSCW '04*. ACM Press, Chicago, Illinois, USA, 142–151. DOI: http://dx.doi.org/10.1145/1031607.1031632

[21] Christian Heath and Paul Luff. 1996. Documents and professional practice: "bad" organisational reasons for "good" clinical records. In *Proceedings of the 1996 ACM conference on Computer supported cooperative work - CSCW '96*. ACM Press, Boston, Massachusetts, United States, 354–363. DOI: http://dx.doi.org/10.1145/240080.240342

[22] Hippocrates. *On the Epidemics*. http://classics.mit.edu/Hippocrates/epidemics.html

[23] J. S. Hirsch, J. S. Tanenbaum, S. Lipsky Gorman, C. Liu, E. Schmitz, D. Hashorva, A. Ervits, D. Vawdrey, M. Sturm, and N. Elhadad. 2015. HARVEST, a longitudinal patient record summarizer. *Journal of the American Medical Informatics Association* 22, 2 (2015), 263–274. DOI:http://dx.doi.org/10.1136/amiajnl-2014-002945

[24] Robert E. Hirschtick. 2006. A piece of my mind. Copy-and-paste. *JAMA* 295, 20 (2006), 2335–2336. DOI:http://dx.doi.org/10.1001/jama.295.20.2335

[25] Abigail E. Huang, Michelle R. Hribar, Isaac H. Goldstein, Brad Henriksen, Wei-Chun Lin, and Michael F. Chiang. 2018. Clinical Documentation in Electronic Health Record Systems: Analysis of Similarity in Progress Notes from Consecutive Outpatient Ophthalmology Encounters. In *AMIA Annual Symposium proceedings.* American Medical Informatics Association, 1310–1318.

[26] Stephen B. Johnson, Suzanne Bakken, Daniel Dine, Sookyung Hyun, Eneida Mendonca, Frances Morrison, Tiffani Bright, Tielman Van Vleck, Jesse Wrenn, and Peter Stetson. 2008. An electronic health record based on structured narrative. *Journal of the American Medical Informatics Association: JAMIA* 15, 1 (2008), 54–64. DOI:http://dx.doi.org/10.1197/jamia.M2131

[27] Daniel Kahn, Elizabeth Stewart, Mark Duncan, Edward Lee, Wendy Simon, Clement Lee, Jodi Frieman, Hilary Mosher, Katherine Harris, John Bell, Bradley Sharpe, and Neveen El-Farra. 2018. A Prescription for Note Bloat: An Effective Progress Note Template. *Journal of Hospital Medicine* 13, 6 (2018), 378–382. DOI: http://dx.doi.org/10.12788/jhm.2898

[28] Mary Beth Kery, Marissa Radensky, Mahima Arya, Bonnie E. John, and Brad A. Myers. 2018. The Story in the Notebook: Exploratory Data Science using a Literate Programming Tool. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press, Montreal QC, Canada, 1–11. DOI: http://dx.doi.org/10.1145/3173574.3173748

[29] Andrew J. Ko, Brad Myers, Mary Beth Rosson, Gregg Rothermel, Mary Shaw, Susan Wiedenbeck, Robin Abraham, Laura Beckwith, Alan Blackwell, Margaret Burnett, Martin Erwig, Chris Scaffidi, Joseph Lawrance, and Henry Lieberman. 2011. The state of the art in end-user software engineering. *Comput. Surveys* 43, 3 (2011), 1–44. DOI: http://dx.doi.org/10.1145/1922649.1922658

[30] Ko, Andrew J., B.A. Myers, and H.H. Aung. 2004. Six Learning Barriers in End-User Programming Systems. In *2004 IEEE Symposium on Visual Languages - Human Centric Computing*. IEEE, Rome, 199–206. DOI: http://dx.doi.org/10.1109/VLHCC.2004.47

[31] Thomson Kuhn, Peter Basch, Michael Barr, Thomas Yackel, and Medical Informatics Committee of the American College of Physicians. 2015. Clinical documentation in the 21st century: executive summary of a policy position paper from the American College of Physicians. *Annals of Internal Medicine* 162, 4 (2015), 301–303. DOI:http://dx.doi.org/10.7326/M14-2128

[32] James Lin, Jeffrey Wong, Jeffrey Nichols, Allen Cypher, and Tessa A. Lau. 2008. End-user programming of mashups with vegemite. In *Proceedingsc of the 13th international conference on Intelligent user interfaces - IUI '09*. ACM Press, Sanibel Island, Florida, USA, 97–106. DOI: http://dx.doi.org/10.1145/1502650.1502667

[33] Christopher A. Longhurst, Taylor Davis, Amy Maneker, H. C. Eschenroeder, Rachel Dunscombe, George Reynolds, Brian Clay, Thomas Moran, David B. Graham, Shannon M. Dean, Julia Adler-Milstein, and Arch Collaborative. 2019. Local Investment in Training Drives Electronic Health Record User Satisfaction. *Applied Clinical Informatics* 10, 2 (2019), 331–335. DOI:http://dx.doi.org/10.1055/s-0039-1688753

[34] Anil N. Makam, Holly J. Lanham, Kim Batchelor, Lipika Samal, Brett Moran, Temple Howell-Stampley, Lynne Kirk, Manjula Cherukuri, Noel Santini, Luci K. Leykum, and Ethan A. Halm. 2013. Use and satisfaction with key functions of a common commercial electronic health record: a survey of primary care providers. *BMC medical informatics and decision making* 13 (Aug. 2013), 86. DOI: http://dx.doi.org/10.1186/1472-6947-13-86

[35] Lena Mamykina, David K Vawdrey, Peter D Stetson, Kai Zheng, and George Hripcsak. 2012. Clinical documentation: composition or synthesis? *Journal of the American Medical Informatics Association* 19, 6 (2012), 1025–1031. DOI: http://dx.doi.org/10.1136/amiajnl-2012-000901

[36] Centers for Medicare and Medicaid Services. 1997. Documentation guidelines for evaluation and management services. *Retrieved August 20 2019* (1997).

[37] Brad A Myers, Andrew J Ko, and Margaret M Burnett. 2006. Invited research overview: end-user programming. In *CHI'06 extended abstracts on Human factors in computing systems*. ACM, 75–80.

[38] Bonnie A. Nardi. 1993. *A small matter of programming: perspectives on end user computing*. MIT Press, Cambridge, MA.

[39] Trygve Nissen and Rolf Wynn. 2014. The history of the case report: a selective review. *JRSM Open* 5, 4 (2014), 205427041452341. DOI: http://dx.doi.org/10.1177/2054270414523410

[40] Sun Young Park, So Young Lee, and Yunan Chen. 2012. The effects of EMR deployment on doctors' work practices: A qualitative study in the emergency department of a teaching hospital. *International Journal of Medical Informatics* 81, 3 (2012), 204–217. DOI: http://dx.doi.org/10.1016/j.ijmedinf.2011.12.001

[41] Sun Young Park, Katie Pine, and Yunan Chen. 2013. Local-universality: designing EMR to support localized informal documentation practices. In *Proceedings of the 2013 conference on Computer supported cooperative work - CSCW '13*. ACM Press, San Antonio, Texas, USA, 55–66. DOI: http://dx.doi.org/10.1145/2441776.2441786

[42] T. H Payne, A. E tenBroek, G. S Fletcher, and M. C Labuguen. 2010. Transition from paper to electronic inpatient physician notes. *Journal of the American Medical Informatics Association* 17, 1 (2010), 108–111. DOI:http://dx.doi.org/10.1197/jamia.M3173

[43] C. Plaisant, R. Mushlin, A. Snyder, J. Li, D. Heller, and B. Shneiderman. 1998. LifeLines: using visualization to enhance navigation and analysis of patient records. In *Proceedings. AMIA Symposium*. American Medical Informatics Association, 76–80.

[44] L. Poissant. 2005. The Impact of Electronic Health Records on Time Efficiency of Physicians and Nurses: A Systematic Review. *Journal of the American Medical Informatics Association* 12, 5 (2005), 505–516. DOI: http://dx.doi.org/10.1197/jamia.M1700

[45] Wanda Pratt, Madhu C. Reddy, David W. McDonald, Peter Tarczy-Hornoch, and John H. Gennari. 2004. Incorporating ideas from computer-supported cooperative work. *Journal of Biomedical Informatics* 37, 2 (2004), 128–137. DOI: http://dx.doi.org/10.1016/j.jbi.2004.04.001

[46] Steven P Reiss. 2019. IoT end user programming models. In *Proceedings of the 1st International Workshop on Software Engineering Research & Practices for the Internet of Things*. IEEE Press, 1–8. DOI:http://dx.doi.org/10.1016/j.future.2013.01.010

[47] S. Trent Rosenbloom, Joshua C. Denny, Hua Xu, Nancy Lorenzi, William W. Stead, and Kevin B. Johnson. 2011. Data from clinical notes: a perspective on the tension between structure and flexible documentation. *Journal of the American Medical Informatics Association: JAMIA* 18, 2 (2011), 181–186. DOI: http://dx.doi.org/10.1136/jamia.2010.007237

[48] S. Trent Rosenbloom, Randolph A. Miller, Kevin B. Johnson, Peter L. Elkin, and Steven H. Brown. 2006. Interface terminologies: facilitating direct entry of clinical data into electronic health record systems. *Journal of the American Medical Informatics Association: JAMIA* 13, 3 (2006), 277–288. DOI: http://dx.doi.org/10.1197/jamia.M1957

[49] S. Trent Rosenbloom, William W. Stead, Joshua C. Denny, Dario Giuse, Nancy M. Lorenzi, Steven H. Brown, and Kevin B. Johnson. 2010. Generating Clinical Notes for Electronic Health Record Systems. *Applied Clinical Informatics* 1, 3 (2010), 232–243. DOI: http://dx.doi.org/10.4338/ACI-2010-03-RA-0019

[50] Adam Rule, Steven Rick, Michael Chiu, Phillip Rios, Shazia Ashfaq, Alan Calvitti, Wesley Chan, Nadir Weibel, and Zia Agha. 2015. Validating free-text order entry for a note-centric EHR. In *AMIA Annual Symposium proceedings.* American Medical Informatics Association, 1103–1110.

[51] Adam Rule, Aurelien Tabard, and James D. Hollan. 2018. Exploration and Explanation in Computational Notebooks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press, Montreal QC, Canada, 1–12. DOI: http://dx.doi.org/10.1145/3173574.3173606

[52] J. L. Schnipper, J. A. Linder, M. B. Palchuk, J. S. Einbinder, Q. Li, A. Postilnik, and B. Middleton. 2008. "Smart Forms" in an Electronic Medical Record: Documentation-based Clinical Decision Support to Improve Disease Management. *Journal of the American Medical Informatics Association* 15, 4 (2008), 513–523. DOI:http://dx.doi.org/10.1197/jamia.M2501

[53] Eugenia L. Siegler and Ronald Adelman. 2009. Copy and Paste: A Remediable Hazard of Electronic Health Records. *The American Journal of Medicine* 122, 6 (2009), 495–496. DOI: http://dx.doi.org/10.1016/j.amjmed.2009.02.010

[54] Ming Tai-Seale, Cliff W. Olson, Jinnan Li, Albert S. Chan, Criss Morikawa, Meg Durbin, Wei Wang, and Harold S. Luft. 2017. Electronic Health Record Logs Indicate That Physicians Split Time Evenly Between Seeing Patients And Desktop Medicine. *Health Affairs (Project Hope)* 36, 4 (2017), 655–662. DOI: http://dx.doi.org/10.1377/hlthaff.2016.0811

[55] Amy Tsou, Christoph Lehmann, Jeremy Michel, Ronni Solomon, Lorraine Possanza, and Tejal Gandhi. 2017. Safe Practices for Copy and Paste in the EHR: Systematic Review, Recommendations, and Novel Model for Health IT Collaboration. *Applied Clinical Informatics* 26, 01 (2017), 12–34. DOI: http://dx.doi.org/10.4338/ACI-2016-09-R-0150

[56] Blase Ur, Melwyn Pak Yong Ho, Stephen Brawner, Jiyun Lee, Sarah Mennicken, Noah Picard, Diane Schulze, and Michael L. Littman. 2016. Trigger-Action Programming in the Wild: An Analysis of 200,000 IFTTT Recipes. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems -*

CHI '16. ACM Press, Santa Clara, California, USA, 3227–3231. DOI: http://dx.doi.org/10.1145/2858036.2858556

[57] David K. Vawdrey. 2008. Assessing usage patterns of electronic clinical documentation templates. In *AMIA Annual Symposium proceedings.* American Medical Informatics Association, 758–762.

[58] Michael D. Wang, Raman Khanna, and Nader Najafi. 2017. Characterizing the Source of Text in Electronic Health Record Progress Notes. *JAMA Internal Medicine* 177, 8 (2017), 1212. DOI: http://dx.doi.org/10.1001/jamainternmed.2017.1548

[59] Justin M. Weis and Paul C. Levy. 2014. Copy, Paste, and Cloned Notes in Electronic Health Records. *Chest* 145, 3 (2014), 632–638. DOI: http://dx.doi.org/10.1378/chest.13-0886

[60] Adam B. Wilcox, Scott P. Narus, and Watson A. Bowes. 2002. Using natural language processing to analyze physician modifications to data entry templates. In *AMIA Annual Symposium proceedings.* American Medical Informatics Association, 899–903.

[61] Lauren Wilcox, Jie Lu, Jennifer Lai, Steven Feiner, and Desmond Jordan. 2010. Physician-Driven Management of Patient Progress Notes in an Intensive Care Unit. *Proceedings of the SIGCHI conference on human factors in computing systems. CHI Conference* 2010 (2010), 1879–1888. DOI: http://dx.doi.org/10.1145/1753326.1753609

[62] Jeffrey Wong and Jason I. Hong. 2007. Making mashups with marmite: towards end-user programming for the web. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07*. ACM Press, San Jose, California, USA, 1435–1444. DOI: http://dx.doi.org/10.1145/1240624.1240842

[63] Jo Wood, Alexander Kachkaev, and Jason Dykes. 2019. Design Exposition with Literate Visualization. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2019), 759–768. DOI: http://dx.doi.org/10.1109/TVCG.2018.2864836

[64] Nan Zang and Mary Beth Rosson. 2008. What's in a mashup? And why? Studying the perceptions of web-active end users. In *2008 IEEE Symposium on Visual Languages and Human-Centric Computing*. IEEE, Herrsching am Ammersee, 31–38. DOI: http://dx.doi.org/10.1109/VLHCC.2008.4639055

[65] Xiaomu Zhou, Mark Ackerman, and Kai Zheng. 2011. CPOE workarounds, boundary objects, and assemblages. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*. ACM Press, Vancouver, BC, Canada, 3353–3362. DOI: http://dx.doi.org/10.1145/1978942.1979439

[66] Xiaomu Zhou, Mark S. Ackerman, and Kai Zheng. 2009. *I just don't know why it's gone*: maintaining informal information use in inpatient care. In *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*. ACM Press, Boston, MA, USA, 2061–2070. DOI: http://dx.doi.org/10.1145/1518701.1519014