RC24656 (W0810-003) October 1, 2008
Computer Science

IBM Research Report

activeNotes: Computer-Assisted Creation of Patient Progress Notes in a Hospital Environment

Lauren Wilcox[1], Jie Lu[2], Jennifer Lai[2], Steve Feiner[1], Desmond Jordan[1]

[1]Columbia University

[2]IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

activeNotes: Computer-Assisted Creation of Patient Progress Notes in a Hospital Environment

ABSTRACT
We present activeNotes, a note creation prototype for use by physicians in a hospital intensive care unit. The application supports the creation of Critical Care Notes, which document the patient's progress and prognosis. We integrate automated, context-sensitive patient data retrieval, and user control of automated data updates and alerts through the use of tags, into the note creation process, without significantly altering the interface to which physicians are accustomed. We performed a qualitative study of the prototype with 15 physicians at New York Presbyterian Hospital. Physicians found activeNotes to be valuable and said they would use it to create both formal notes for medical records and informal notes. We were surprised to find that while physicians have rejected template-based systems in the past, they expressed a desire to use activeNotes to create personalized, doctor-specific note templates to be reused with a given patient, or for a given condition.

Author Keywords
Medical note creation, tags, clinical notes, clinical documentation, medical user interfaces, information retrieval

ACM Classification Keywords
H5.2 [Information Interfaces]: User Interfaces, Input. I.3.6 [Methodology and techniques]: Interaction Techniques.

INTRODUCTION
A patient progress note is a clinical document, written daily by a hospital physician, describing a patient's status and the physician's assessments and care plans for the patient. An attending physician, has primary responsibility for the patient's care, composes a patient progress note, called the *Attending Critical Care Note,* for each of her patients. These patient progress notes are referred to by other physicians caring for the patient, and are included in the official medical record for legal and billing purposes.

Creating a Critical Care note requires a physician to gather, review, and comment on previous and current patient data such as lab results, information from medical rounds, medications, procedures, and tests to determine patient health, as well as select relevant information to put into the current note. Current Electronic Medical Record (EMR) systems [10] include facilities that can assist note creation; however, the complexity of their user interfaces has been cited as a key barrier to their adoption by physicians [7]. For example, many EMR systems have multiple patient information windows, as well as deeply structured menus and data pathways, making the daily creation of patient notes more difficult and time-consuming than it needs to be.

As a result, many physicians retrieve patient data manually from an EMR system or a database, record device readings, and receive oral briefings by residents, fellows and nurses. They then use generic document processing systems, such as Microsoft Word, and employ basic text editing techniques such as typing, copying and pasting to insert relevant patient data into a progress note document. This data is static note content that requires physicians to manually update the note if the values change from the time it is first created to the moment it is printed and inserted into the patient's record. If the data that needs to be included in a note is not available at the time the note is created, physicians must remember to include the data at a later time when it becomes available. This is not uncommon since attending physicians often request the information from residents, who may not have the data when requested. The physician must then rely on their memory, or the resident to make sure it ultimately gets included in the note. A survey we conducted of physicians in two Intensive Care Units (ICUs) at New York Presbyterian Hospital (NYPH) indicates that the current note creation practice is time-consuming and error-prone, due to its reliance on manual input and susceptibility to typographical errors and omissions.

To address the shortcomings of their current note creation system, we are developing *activeNotes*, a medical note creation prototype to explore ideas around assisting physicians in data retrieval and data updates for the creation of patient progress notes. We introduce *activeTags* to support user control of updates to patient information that is inserted into a note from dynamic data sources (e.g. patient database, or lab systems). We also explore the specification of user-customized alerts associated with these updates.

ActiveNotes is an integrated environment that offers physicians two side-by-side views (see Figure 1): an editable note view and a patient information view in which the system displays results from data queries. As a note is being edited, activeNotes dynamically interprets new content created by the physician in the context of the existing note to detect potential information requests. If requested via a hot-key, the system automatically formulates the corresponding queries for retrieving information from multiple data sources. The physician can review and insert the retrieved data in real-time, as well as associate an *activeTag* with note content that will control subsequent updates to that data. Each activeTag links the tagged content with the automatically-generated queries and data actions for retrieval, updates, and alerts. The physician can configure the actions of an activeTag to obtain the updated values at specified times, and have these updates automatically reflected in the note, as well as evaluated against user specified alert mechanisms.

In the following sections, we first describe related work and contrast it with our approach. We then present insights about physicians' workflow and current process for note creation gained from observations, semi-structured interviews and a survey with 12 physicians (four residents and 8 attending physicians) in two NYPH ICUs. Next, we describe the activeNotes prototype, whose design was informed by this work. Finally, we present findings and feedback from a qualitative study of the prototype conducted at NYPH with 15 physicians, and conclude with a discussion of the findings and potential future work.

RELATED WORK

The effectiveness of an integrated environment for searching, reading, and creating notes with the goal of sense making in the medical domain has been explored in systems such as Entity Workspace [1]. It addresses note creation tasks with the purpose of discovering new high-level information from structured content (e.g., question answering). It includes automatic highlighting of terms, techniques for importing text from documents into a note, and support for annotating and organizing information in a note. While we also provide users with an integrated environment for searching over documents and creating content, our focus is on supporting specific queries to retrieve relevant information from dynamic data sources as well as previous patient notes, for the patient for whom a note is being created.

The term "tag" is often applied to a text string used to group items. For example, there are tagging applications that allow a user to tag web pages, photos, and multimedia clips with terms, as well as perform search and group operations based on these terms [e.g., 3, 8]. While we are inspired by these types of tags, we extend the idea of a static tag used to group or describe items. ActiveTags support tagging dynamic data entries, and tags for these entries serve as identifiers of the data, placeholders that reflect the ultimate values of the data, and a set of rules to control how the data entries are reflected in a document.

The idea of including instances of document fragments from different sources directly into a destination document that provides "windows" into the original document fragments has been termed *transclusion* by Nelson [9]. Whenever the source document content fragments are updated, the embedded reference in the new document reflects these updates. Building on this idea, we wish to provide users with mechanisms to manage how dynamic source content is reflected in the new document. Our source content is determined by the interpretation of an information request made by a user thta automatically formulates queries for searching for patient data from multiple sources.

Smart Tags, a facility incorporated into Microsoft Office products, can automatically recognize common entity types such as a person's name or address, and support type-specific actions to perform common tasks (e.g., add a name to an Outlook address book) [6]. A Smart Tag can also be pre-configured to link to content (e.g., a legal clause) in a content management system, such that changes to text in the content management will dynamically be populated to the linked content using the Smart Tag. ActiveTags differ from Smart Tags in three key respects. First, upon creation of an activeTag, our system interprets the content it is associated with in the *context* of other text in the document in order to formulate queries on source content. For example if the query needs identifying patient data for the query, it will obtain it automatically from other sections of the Note. Second, activeTags provide users the freedom to determine what to tag and offer control of update and alert mechanisms for managing the tagged content. Third, rather than linking to a specific single source, activeTags are be associated with one or more queries, such that the content linked to by an activeTag is not a document, single entry in a database, or action, but a set of queries that may be used to retrieve results according to user-specified, data-aware, rules.

Our use of activeTags to assist note creation is inspired by the work of Hsieh, Lai, Hudson and Kraut [4]. They introduce tags in instant messaging (IM) that alter the behavior of the tagged items (messages) to facilitate near-synchronous communication in IM clients. Senders can tag their IM messages to trigger different types of support on the receiver's side for different types of tasks (e.g., tasks that do not require immediate attention, or tasks that have deadlines).

NOTE CREATION IN THE ICU

The design of activeNotes was informed by our observations of six physicians' workflow, environment and note creation strategies over an elapsed time period of six months in the NYPH Cardiothoracic ICU (CTICU), including observations and semi-structured interviews. We also conducted a survey of eight attending physicians in the CTICU and Surgical ICU (SICU). Below we summarize these findings.

Current EMR systems deployed in the NYPH ICUs use either form- or template-based interfaces for note creation. For example, Eclipsys Sunrise Clinical Manager™ provides pre-configured documentation templates, configurable at the administration level rather than the physician level [2]. Physician must spend time navigating through hierarchical structures to examine data, instead of viewing it in the context of the note they are creating. VISICU *e*ICU® eCareManager [12], and Web-based Clinical Information System (WebCIS) [5, 11] provide form fill-in user interfaces with limited pull-down choice lists for data and annotations as well as text boxes that physicians must fill in individually.

While these current systems allow physicians to create notes that can be easily translated to appropriate billing codes required for billing and legal record keeping, physicians find the interfaces to be too restrictive for three main reasons. First, physicians write about half of a patient note using free-text entry (the patient's past history, current diagnosis, reactions to certain treatments or tests, and assessments and plans); however, current EMR systems do not allow physicians to adequately enter free text. Second, using these systems, physicians are forced to document various aspects of a patient's health in a pre-set order, which makes editing, review, and analysis of patient data at different stages of note creation difficult. Third, in an ICU, the note structure and relevant data items for a patient may change depending on what goes wrong with this patient and which organ systems are affected; therefore, it is difficult to capture and express the clinical condition of a patient for the different types of possible scenarios with any one standard form or template. A representative comment from physicians about templates is: "I am hostile to rigid templates. They impede my ability to think about the patient. I make many connections and they don't represent how I think". As a result, most physicians continue to use a regular document editor, since its familiar interface gives them complete control over what to include in a patient note and where to put it.

To create a patient note and to formulate the assessments and care plans for the patient, physicians gather factual patient information from multiple sources such as the EMR systems, the patient database, printed lab reports, prior patient notes, and oral presentations or written records of residents and fellows. Their information needs for note creation are dynamic and context-sensitive, as they are highly dependent on patient status and any content that has already been entered in the patient note. For example, below the heading "Abdomen" in a note section entitled "24 Hour Events", physicians may need lab results for the past 24 hours related to the patient's liver function. In contrast, below the same heading "Abdomen" under the "Physical Exam" section of a different note, physicians may need information about whether or not bowel sounds were present for the patient during the most recent physical exam. After reviewing and determining the relevancy of the gathered data, physicians manually insert relevant patient information in the patient note.

Summary of Attending Physician Survey

Among the eight attending physicians surveyed, four have been working in an ICU for less than three years, one for five years, and three for more than 20 years. They estimated that a typical day for them in an ICU lasts around 9–12 hours, during which they reported spending 5 hours on average (minimum 2.5 hours, maximum 8 hours) on medical rounds for patient care. Each physician estimates writing between 10–18 (mean = 16) Attending Critical Care Notes per day. Six create 80–90% of note content during medical rounds, while two create their notes after rounds, relying on their memory.

Five of the six attending physicians who compose patient notes during medical rounds at the patients' bedside use a laptop computer and a document processing application such as Microsoft Word. One physician hand-writes patient notes during rounds and types them into a computer afterwards. All physicians we surveyed consider the task of collecting relevant and correct patient data the greatest challenge in composing an Attending Critical Care Note. They admitted spending a considerable amount of time on navigating through previous notes to locate relevant patient information, especially notes written by other physicians.

A patient note is usually not printed and inserted into the patient's record immediately after it is created. The physicians estimated that as many as eight hours could elapse between the point at which the note is created and the time it is filed, during which they continue monitoring patient status. Throughout the day, the physicians keep track of patient information, such as lab results, vital signs, and ventilator settings, to analyze a trend of measurements, detect abnormalities, and adjust assessments and plans for patient care accordingly. While the attending physicians all agree that patient notes should be updated to reflect the above changes, they have different opinions on when the updates should be performed. Two physicians think notes should be updated immediately when new information becomes available; two would like to update notes periodically; four consider it sufficient to perform updates once before notes are printed out and put into medical records.

When asked how convenient it is to make updates to an Attending Critical Care Note directly using the current note creation systems, six of the eight physicians said that it was

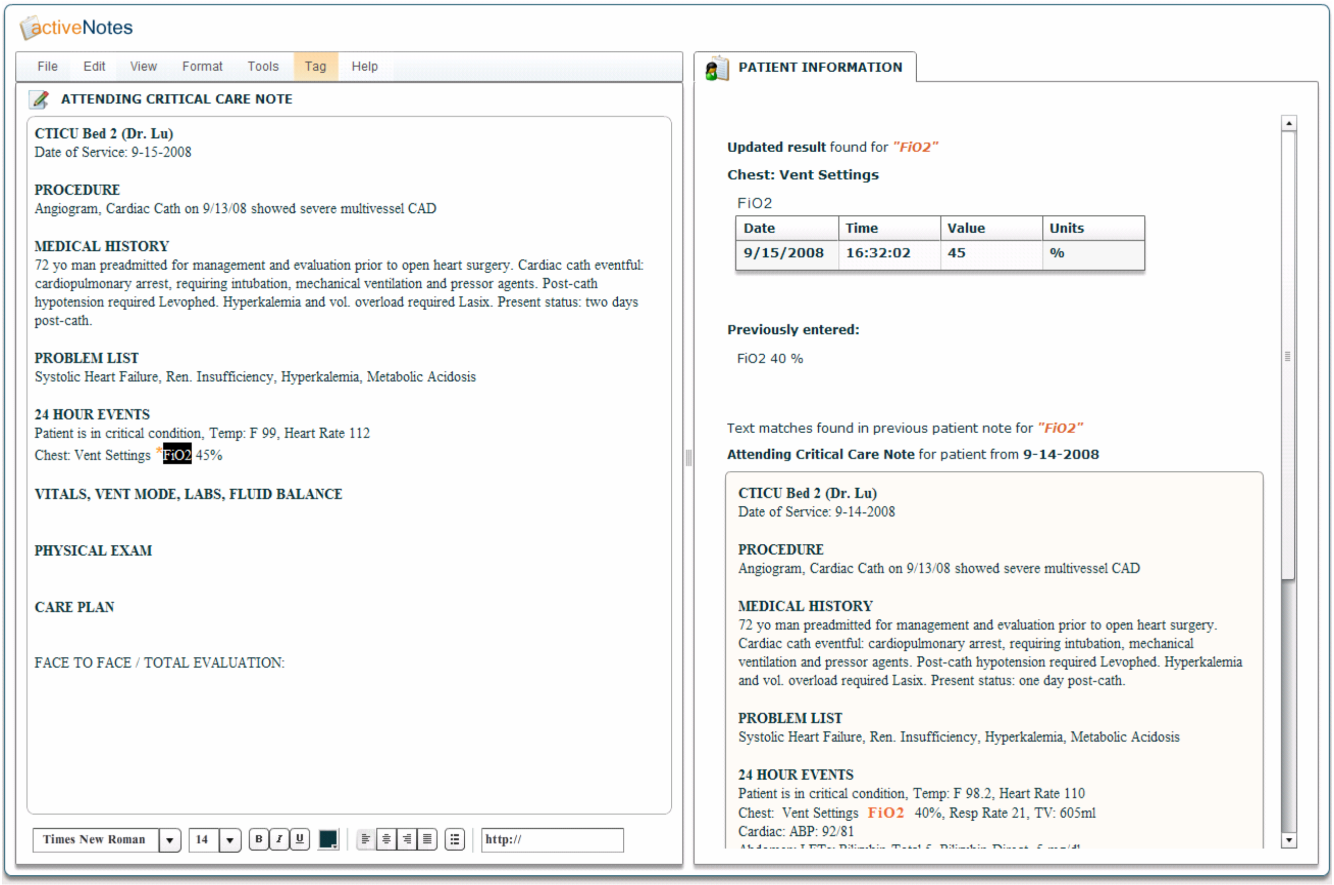

Figure 1. activeNotes delivers an update to the note. The note appears on the left side of the screen, and the retrieval results of updated patient information appear on the right.

either somewhat inconvenient or very inconvenient. They largely rely on their own memory or brief reminders jotted on paper to remember the specific items that need to be updated in a note, and such a list can vary from patient to patient. One physician chooses instead to fill out an entirely new note if he feels an update is worth noting. Follow-up by residents is the primary source of the updated information, and the physicians have to manually edit each note once they obtain the updates from residents.

In summary, the results of our studies and survey indicate that due to limited system support for data retrieval and updates, physicians spend a substantial amount of effort composing notes and keeping notes updated, leaving them less time for patient care.

ACTIVENOTES

In this section we present the design and implementation of activeNotes, a medical note creation prototype created in response to the insights gained during our preliminary fieldwork with physicians in the ICUs.

System Design

A key goal of our design is to enable user-controlled patient data retrieval and automated updates within a word processing interface with which physicians are familiar and comfortable. Another key goal is easy reference to and navigation of the previous day's note, which we found to be referenced frequently during note creation.

Data Retrieval

Data retrieval in activeNotes is supported by the recognition of text in the context of the note entered in the note view (Figure 1, left). When requested, the system looks at text that the user has just typed, highlights the last term that it recognizes as an information request, and automatically formulates queries to retrieve information relevant to the request from appropriate data sources. Each information request is interpreted in the context of the existing note so that relevant information (e.g., patient identity, date, time, or the organ system under review) can be embedded by activeNotes in the automatically generated queries. Users can request a single piece of information (e.g., heart rate), or multiple pieces of related information at once (e.g., ventilator settings, information at the organ system level such as renal). Retrieved information is placed in the patient information view (Figure 1, right) and can be automatically inserted into the note. In this way, note-driven retrieval allows users to dynamically gather data while entering free text and without leaving the current interface or losing control over content, format, or structure.

As an example, consider an attending physician, Dr. Smith, who is using activeNotes to create a note for a patient. To review the patient's current status, Dr. Smith wants to check on the patient's *vent settings* (the settings of mechanical ventilation equipment used to replace natural breathing for a patient that cannot maintain normal respiratory function). A patient receiving mechanical ventilation is often *intubated*, meaning that a tube inserted into the patient's airway, rather than a mask worn on his face, delivers oxygen.

Using activeNotes, Dr. Smith types "vent settings" in the note and presses Ctrl-Space to request relevant data. The system detects the information request and formulates database queries to retrieve the values of relevant data items such as the *FiO2* setting (fraction of inspired oxygen, or percentage of oxygen of each "breath" that the ventilator provides) and *tidal volume* (the volume of air of each "breath" that the ventilator provides) from the patient database. At the same time, it also formulates a keyword-based query to search for relevant content from yesterday's note for the patient. If Dr. Smith does not need multiple ventilator settings at once, she can explicitly request something more specific (e.g., "FiO2"). The numeric data items retrieved from the database are presented in interactive charts or tables, the format being determined by the amount of data retrieved and user-set preferences. The previous patient note is also displayed with the matched keywords highlighted. The physician can click on the data she deems relevant to the assessments and plans for the patient, causing it to be inserted into the note automatically at the position where she issued the information request.

Data Updates

An activeTag is an annotation that is attached to a content fragment and associated with data actions (retrieval, updates, and alerts) that act upon the tagged content. Users can attach activeTags to data-related note content to indicate their wishes to obtain live updates, or to receive alerts when the automatically updated content meets certain criteria (e.g., exceeds a threshold). Users can also use activeTags to request automated updates for patient data that is not yet available when initially requested. This way, users can avoid forgetting to revisit a patient note to fill in the missing data.

Users can configure an activeTag by choosing among different options for when and how to perform updates. For example, a user can request that an update be run immediately, at a specific time (e.g., 5 pm), or at a specified schedule (e.g., every 10 minutes). Users can specify through preference options whether or not the originally inserted value be automatically replaced with the updated value.

In addition to update options, users can request that an alert be generated if user-specified criteria are met. For example, a user can elect to receive an email or SMS message when the updated value goes over (or under) a threshold, or changes (increasing or decreasing) by a specified amount.

Besides update and alert tags, physicians can create tags with labels that are meaningful to them, to be able to easily organize and update content across patient notes. In this way, the tags are used in the more traditional manner, but are still "active". At any time, a user can choose to view and manage all the activeTags organized by labels, or based on user-specified update or alert options (e.g., times or frequencies of updates, and types of alerts). Other benefits offered by activeTags include the ability for users to easily track the value of a data item over time and organize follow-up tasks.

ActiveTags not only enable easier and more efficient data updates of existing patient notes, but also provide users the ability to generate an "active" note template with user-configurable data actions that can be reused for automatically populating data-related portions of later patient notes. Such a note template can significantly expedite note creation and allow users to focus on formulating assessments and plans for patient care instead of data retrieval and updates.

System Implementation

We implemented activeNotes using a combination of Adobe Flash with Adobe Flex 3 for the user interface and Java for the back-end. In this section, we first describe the user interface of activeNotes, followed by its architecture and implementation.

User Interface

As shown in Figure 1, the user interface of activeNotes includes a set of menu items and two main interaction areas: the Note Area on the left, entitled "Attending Critical Care Note" and the Results Area on the right, entitled

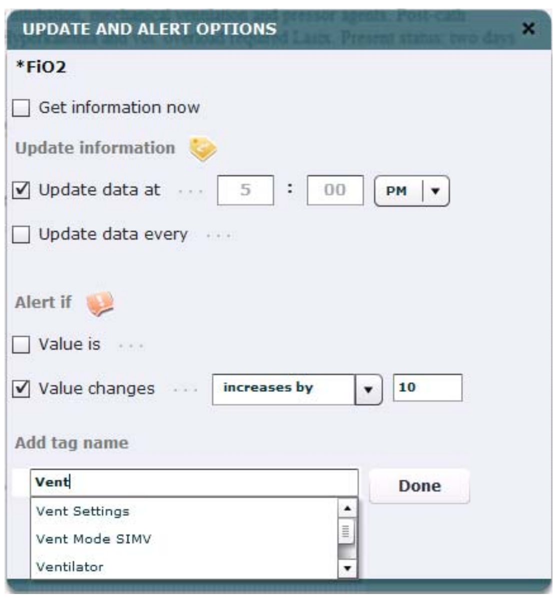

Figure 2. activeTag menu.

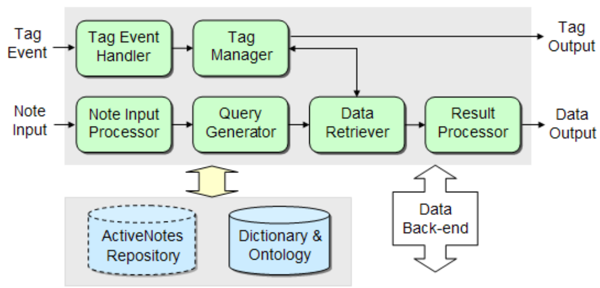

Figure 3. activeNotes system architecture.

"Patient Information". The menu items include File operations that allow a user to create, open, or save a note file, and Tag controls and preferences that support viewing, organizing, and managing activeTags, and allow users to set their preferences for updates and alerts (e.g., how the updated results or the alerts should be delivered). All activeTags are automatically saved and embedded in the notes in which they are defined, so that opening a previously created note reloads its activeTags as well.

The note area has the look and functionality of a rich text editor. A user can type her note as she normally would, and at any time during note editing, can signal the system (by pressing Ctrl-Space) to retrieve the needed patient information based on the content inserted into the note thus far.

The result area displays the retrieval results for each information request. The numeric results are presented in charts or tables followed by the previous patient note with highlights to indicate the content that matches the keywords derived from the request. A user can click a data point in a chart or anywhere in a row of a result table to indicate her wish to have the corresponding result automatically inserted into the current note. Content from the previous patient note can be copied and pasted anywhere in the note area.

To associate an activeTag with some content in the note, a user can click anywhere within the word to have it selected and highlighted, and right click the mouse to bring up the context-sensitive Tag Menu (Figure 2). The tag menu displays the content being tagged, and provides a number of options for retrieval, updates, and alerts. The option to "*Get information now*" is the default action, and it is what the system does if the user just activates an update (Ctrl-Space) without bringing up the tag menu. The option "*Update data at …*" allows users to schedule an update at a particular time, while the option "*Update data every …*" is used to schedule periodic updates at regular intervals. For alerts, users can choose to receive them when the updated value is above/below a particular value, and/or when the updated value increases/decreases by a specified amount relative to the original value. When a user finishes configuring an activeTag and clicks "*Done*" to close the tag menu, an orange icon appears to the immediate left of the tagged content in the note area to indicate that an activeTag has been created for that content. Double-clicking on the indicator of an activeTag activates its tag menu again, so users can modify previously specified options. To remove an activeTag, a user can simply delete its indicator.

System Architecture

The back-end system of activeNotes, shown in Figure 3, consists of a number of domain-independent modules and a set of system-maintained resources. The resources include a domain-dependent *ontology* that stores semantic elements and their relations, a *dictionary* that maps words to these semantic elements, and the *activeNotes repository*, which records all existing note content, information requests, queries, results, tags, and the relations among them.

When a user creates new note input and signals the system for data retrieval, it is passed to the *note input processor* module, which detects the information requests specified based on the dictionary. Specifically, the system adopts an existing natural language input processing algorithm [13] to identify a set of semantic elements from the input (e.g., data concepts and attributes) by matching the input with dictionary entries. It also infers the relationship between these semantic elements based on the data ontology. The output of this algorithm is a set of information requests encoding the semantics of the detected user information needs. The module then uses the existing note content stored in the activeNotes repository to interpret each detected information request in context. Next, for each information request, the *query generator* module derives a set of query parameters for each data source based on the dictionary, the data ontology, and the mapping between semantic elements and database tables/columns. System-executable queries are generated by assembling the query parameters using source-dependent procedural methods. The *data retriever* module connects to an application back-end for retrieving from an external data source. After the queries are executed, the retrieval results are passed to the *result processor* module, which determines how to visualize the results (e.g., chart or table) and generates the proper formats that can be interpreted by the interface for correct rendering.

When a user performs a tag-related interaction (e.g., creates a new activeTag), the corresponding event is sent to the *tag event handler* module. This module extracts the parameters of options and operations, and passes them to the *tag manager* module. Based on the parameters, the tag manager fetches resources from the activeNotes repository to perform the needed operations. For example, to create a new activeTag, the tag manager finds in the activeNotes repository the note content being tagged, retrieves its corresponding information request and queries, and links them with the tag. The tag manager also configures the tag's data actions based on the parameters of the user-specified options (e.g., setting timers for periodic updates), and initializes the alert mechanisms if requested. To perform automatic updates, the tag manager monitors the scheduled updates for all activeTags and contacts the data retriever for obtaining the updated results. In addition, the tag manager checks whether alerts should be triggered by the updated results and creates alerts if true.

QUALITATIVE STUDY OF ACTIVENOTES

We conducted a qualitative study with 15 (11 male, 4 female) physicians at NYPH, all between 29–55 years old; 11 are currently attending physicians, and 4 are residents with one to four months of experience in an ICU.

Environment

Working within the realities of the hospital ICU posed some challenges for the design of our study. While physicians were agreeable and generous with their time, most are on call while at the hospital, and a request for even 30 minutes of their time is a lot. Thus we needed to plan a training session, task, and survey that could be completed in no more than 30 minutes. Since we were at risk of interruptions from cell phones and pagers during that time, we opted for qualitative feedback during and after use of the system rather than quantitative timing metrics to compare throughput with their current process.

Method

Both the training and study task were performed using a laptop computer we provided with a mouse that could optionally be used instead of the built-in trackpad or trackpoint. The task involved first reading a scenario setting the background information on a fictional patient, and two Attending Critical Care Notes of this patient from the previous day. The scenario and the previous patient notes were based on a randomly selected, anonymized patient profile from historical data that we have. Of the two Attending Critical Care Notes provided for training purposes, one resembled a standard note printed for the patient's medical record, with no additional annotations. The other was annotated to include underlined and emboldened terms. These annotated terms indicated words that the system had recognized and used to retrieve patient data results. Figure [4] shows a fragment from the annotated note.

…

PROBLEM LIST

Systolic Heart Failure, Renal Insufficiency, Hyperkalemia, Metabolic Acidosis

24 HOUR EVENTS

Patient is in critical condition, **<u>Temp:</u>** F 98.2, **<u>Heart Rate:</u>** 110

<u>Chest:</u> **<u>Vent Settings</u>** **<u>FiO2</u>** 40%, **<u>Resp Rate:</u>** 21, **<u>TV:</u>** 605ml

…

Figure 4. A fragment of the annotated note used for training.

After a participant read the patient scenario, the study coordinator introduced activeNotes comparing and contrasting it with word processing applications with which the participant is familiar, and described the features with examples. Training included using three sample terms for which the system formulated queries and provided results. Results were presented in the right hand panel of the application, with highlighted occurrences of the keyword in the previous patient note, and other data query results. Thus, the participant could also use the information request utility to navigate the previous Attending Critical Care Note as well as view results from the patient database.

In the examples, the study coordinator demonstrated the difference between an information request to the system on a specific item like "FiO2", which is a sub-category of "Vent Settings" and a system call at a higher level, such as "Vent Settings". With the latter the system returned multiple ventilator settings for the patient, including FiO2. The third example was an information request for an even less specific term, "Chest". Results here included tables of data items that would be noted in the Chest section, which included ventilator settings, arterial blood gases and other information related to the patient's respiratory system. In all cases the previous (i.e. yesterday's) note was displayed with the corresponding terms highlighted. The physicians were shown how to insert data by clicking on the results. Finally, they were shown how to tag note content to set automatic updates and create personalized data alerts.

Participants then performed information requests for a necessary information item given a patient with the sample condition. Once they had successfully completed the requests, we asked them to continue completing the Attending Critical Care Note for this patient, allowing them to use the system without intervention. Three sections of the Note were pre-filled-in to provide some context. Physicians were asked to focus on one of the following empty sections: "24 Hour Events" or "Vitals, Vent Mode, Labs and Medications". We asked each participant to speak while using the system, articulating their reasoning process for deciding what to note and commenting on their experience obtaining and inserting data related to their information needs.

Since we had sample data for labs, vital signs, blood gases and ventilator settings, we instructed them to assume that any information they could not look up had not changed from the previous day. They were allowed to refer to the annotated note for examples, as well as enter any terms they wanted information on, even if those terms were not listed as examples on their reference sheet. After completing a section of the note, we asked each participant a series of qualitative questions to structure the feedback elicited from participants. Sample questions included "What is the greatest benefit of the system?", "What is a major drawback of the system?", "In your opinion, would physicians use this? If so, why? If not, why not?".

FEEDBACK

In this section we summarize key preferences, problems, and recommendations distilled from the qualitative user study of activeNotes.

The physicians were uniformly positive in their desire to use activeNotes to compose patient progress notes. Several volunteered that it was an improvement over the current method for retrieving and noting patient information. Sample comments include: “this is head and shoulders above what we’re using now”, and “this is a heck of a lot better than anything else I’ve used”. However one physician believed that typing any part of the Attending Critical Care Note is an administrative task and that any system that required typing was unusable. While he would not complete the task he answered several questions based not on actual use of the system, but on his impressions and opinions given the examples we demonstrated.

With regard to favorite features, half of the participants, most of them attending physicians, explicitly mentioned the ability to tag items for updating and/or alerting as the key feature that they would keep. Most others also mentioned the importance of tagging for either updates or alerts at other points in the survey. Opinions varied as to whether updates or alerts were the more important form of tagging. In all, using tags to set up either updates, alerts, or both were considered important by 13 out of the 15 participants. Of the two who did not consider these tags to be important, one was the physician who would not type and could not conceive of how he would use the tags with a spoken system, and the other was a resident who would prefer to respond to the alerts already in place in the ICU for abnormal values. He also mentioned that he would prefer a system that allows a physician to place orders for medications and tests, and that he would like to set up alerts for the purpose of being notified when a “tagged task” was completed (e.g., after the results of a test he ordered are in the database).

When asked to describe the greatest benefit of our system, physicians offered phrases like “[it is] easier to stay organized about following up on things” and “I like being able to see yesterday’s note like that”. Benefits frequently named included those related to time savings, efficiency, ease of inserting items into the note, ease of updating the note. Physicians felt the facility with which they could include “fresh” information might result in higher quality notes: “What I like about this is that every note that is composed is ‘fresh’, I can bring in today's information easily without having to retype so many things, so I don't worry about copying something and not updating it, but I can also write comments and put things exactly where I want them in the note… When we do include one from the results, it has a value and a unit, and this is good because, we’re told not to write things like, ‘insulin 10 u’. I think this is a good mix. A system like this makes more sense than the alternatives now.”

Also appreciated were the ability to view results including the most recent values with a 24-hour trend, and the ability to set up alerts. Less frequently named benefits included avoidance of transcription errors and ease of referencing the previous note.

Major drawbacks mentioned included a concern that it might take a while to learn what keywords are recognized by the system. While we had prepared a vocabulary based on the corpus of the data available to us, we found that physicians might hit CTRL-Space after patient information headings that made sense, but for which we did not have an entry in our dictionary. For example, one physician typed “GTT” which activeNotes failed to recognize as a request for the current “drips” that the patient was receiving and their rates. Participants were unclear about how to determine whether or not a word or abbreviation that they used in a note would be understood by our system. Some tried up to five unsuccessful information requests before they issued one which yielded expected results. Incorporating new terms and acronyms is a learning aspect of the system, which was far from mature when we obtained feedback on the prototype.

Physicians were also unclear about what types of information the system could provide. Attending physicians were more confused about this than the residents we studied. Attending physicians often obtain status information from the residents, thus some wanted to submit information requests pertaining to patient status, such as “intubated”, whereas residents looked for data categories such as “Liver Function Tests”. One reason for this might be that since residents spend a considerable amount of time retrieving data from hospital information sources, they may be more accustomed to the types of information requests and results we supported and presented within activeNotes.

When asked what additional features would need to be included to make the system more usable and useful, many physicians mentioned that they thought a multi-level completion-style menu presented while typing, which would allow them to view several levels of possible requests that matched their input, would be helpful. For example, typing “Ch” would allow them to navigate to a menu item entitled “Chest”, which could expand a submenu containing items belonging to the “Chest” category such as “Vent Settings”, “Patient Respiratory Rate”, etc. Both top-level menu items and child menu items would consist of system-recognizable information requests.

DISCUSSION

Importance of activeNotes with activeTags

We conclude that the user interface design of activeNotes for creation of patient progress notes is an improvement on current applications used for the creation of medical progress notes for the hospital physicians we studied. We also conclude that activeTags for updates or alerts in an integrated environment for progress note creation and patient data retrieval would be a helpful addition to these physicians. Longer study is needed to compare responses across physicians in other ICU environments and address usage scenarios, system requirements, and deployment details of activeTag controls in a system like activeNotes.

We captured several illustrative responses from physicians when presented with activeTags. As one participant mentioned, “Take cultures, for example. I might only tag culture results for an alert. But this is something I would definitely use. Cultures take three days and it could be easy to forget by then that they need to check for them. But these are absolutely crucial to the diagnosis of infectious diseases. ” Another said “I’d probably tag everything, because I like to stay on top of things in whatever way I can”. Most attending physicians also mentioned that they preferred to have the choice of which items to tag in the note, because residents and nurses typically respond to hospital alerts, then approach the attending with their conclusions. One said, “there are certain things I would want to be alerted about directly, even when I’m away from the hospital.”

When asked how they would like alerts to be delivered if used, almost all of the participants replied that they would want a text message sent to their phone, or text pager. A few preferred instead to see a visual indication in the note. About three mentioned that they would like to configure the alerts for others, including themselves. One said, “Alerts would be really great if I could not only set one up for myself, but for the resident”. He then added, “in fact, sometimes I’d probably set these up to be delivered only to the resident, as a way to remind them to follow up on this thing”.

Semantics of Information Requests

As we conducted our study, we paid attention to how the participants specified their information requests. We were interested to find out if they would specify requests at multiple levels and if so, whether or not a strategy or preference would emerge. All of the participants who used our system showed a strong preference for the ability to specify information requests at the three levels we provided, and their think-aloud feedback helped us define the three levels further:

1. *Information Item* – a request such as “Temp” which yields results related only to the specific item requested.

2. *Information Category* – a request such as “Labs”, “Liver Function Tests” and “Vitals” which yields a related set of information item results. The categories include lab and test results, physical exam results, and life support settings.

3. *Organ*/*Organ System* – a request such as “Renal”, “Endocrine”, “Abdomen”, “Chest”, and “Cardiac”, which yields a combination of results from individual information item requests as well as from related data categories, thus results contain possibly many sets of information item results.

Most physicians used a less specific request first, such as that in the *Organ/Organ System* category, followed by specific requests for additional information, such as those in the *Information Item* category, for each major topic they addressed in the section of the note they completed. Many commented on this strategy, mentioning that they expected this would help them save time typing, while giving them the ability to insert what they thought was necessary, exactly where they wanted it in the note. A request for ‘Chest’ would yield results that the physicians could import into the note, with both the heading and the value of the patient information automatically composed. If the physician wanted to note a related vital sign not included in the results for “Chest”, she could request that vital sign as an *Information Item* request. In contrast, searching for each *Information Item* such as “Temp” meant typing each individual heading and selecting corresponding data from the results for each.

Creating Personal Templates

One surprising finding of our study was an inclination expressed by the physicians to try to use our system to create templates. We had avoided a template-based GUI, based on responses offering editable sections in a rich text editor to match the clean interface of the word processing applications the physicians were accustomed to. When introduced to the automated data retrieval capability and activeTags, half of the attending physicians we studied expressed a desire to use our system to create their own template notes.

These physicians mentioned that they would create sample notes with information requests as “placeholders”. The information requests would be applied to specific problems, or problem combinations. They would apply the sample notes to a patient based on problems the patients were experiencing, then visit each information request, setting up updates to reuse the note the next day with the most up-to-date values already inserted.

One commented, "My notes are in my own format, so I can easily recognize them. I want to create that format myself. I want to be able to do things smoothly, and decide when I put in values that I think are important, not be told what to put in and in what order".

Another said, "Patients have different profiles. For a problem, I’d probably set up a data profile, then set updates according to how important it is to monitor each, for a certain problem. This is the way I'd use updates."

CONCLUSION AND FUTURE WORK

In this paper, we present activeNotes, a note creation prototype that supports the creation of patient progress notes by physicians in a hospital intensive care unit. We have focused on the design and implementation of activeNotes from two key aspects. First, we integrate automated, context-sensitive patient data retrieval into a note creation interface with which physicians are familiar and comfortable. A user can enter free-text information requests in the note she is writing. The system automatically recognizes these requests by interpreting new note input in the context of the existing note, formulates corresponding queries, and retrieves relevant information. Second, we introduce activeTags to support user specification of automated updates and alerts for the relevant patient data in a note. A user can attach activeTags to data-related note content for automated updates and alerts, and configure the tags based on her needs and preferences for these updates and alerts, e.g., when and how to perform updates, when and how to deliver alerts. Our qualitative study with 15 physicians at New York Presbyterian Hospital shows that the new features provided by our system were well-received and considered very beneficial to patient note creation. Physicians were excited about the potential usefulness of using activeNotes to create personalized, doctor-specific note templates to be reused for a given patient or a given condition, which we didn't expect given their rejection towards **prior** template-based note creation systems. This finding points out a promising direction for future development of computer-assisted patient note creation systems.

Currently, activeNotes connects with patient data pulled from a single source. The data has been altered to protect the identity of patients. Our most pressing next step is to add integrated system support for connecting with and retrieving from multiple systems and data sources used in New York Presbyterian Hospital ICUs now, so that our prototype system can be studied in a larger-scale realistic trial for patient note creation tasks.

ACKNOWLEDGMENTS

Left out for anonymous submission.

REFERENCES

1. Billman, D., Bier, E. Medical sensemaking with Entity Workspace. *Proc. CHI 2007*, 229–232.
2. Eclipsys. http://www.eclipsys.com/
3. Flickr. http://www.flickr.com/
4. Hsieh, G., Lai J., Hudson S. and Kraut R. Using tags to assist near-synchronous communication. *Proc. CHI 2008*, 223–226.
5. Hripcsak, G., Cimino, J., and Sengupta, S. WebCIS: Large scale deployment of a Web-based clinical information system. *Proc. Amer. Med. Informatics Assoc. (AMIA) Symp.*, 1999, 804–808.
6. Hughes, G. and Carr, L. Microsoft Smart Tags: Support, ignore or condemn them? *Proc. Hypertext 2002*, 80–81.
7. Miller, R.H., Sim, I. Physicians' use of electronic medical records: Barriers and solutions. *Health Affairs.*
8. Muller, M. Patterns of tag usage across four diverse enterprise tagging services. *Proc. HCIC 2007*
9. Nelson, T. H. (1995). "The Heart of Connection: Hypermedia Unified by Transclusion". *Communic of the ACM* (8): 31–33.
10. Tang, PH (ed). *Key capabilities of an electronic health record system.* Washington DC: Institute of Medicine, National Academy Press, 2003.
11. WebCIS. https://webcis.cpmc.columbia.edu/webcis/
12. VISICU eICU eCareManager. http://www.visicu.com/products/evantagesystem.html/
13. M. Zhou, K. Houck, S. Pan, J. Shaw, V. Aggarwal and Z. Wen. Enabling context-sensitive information seeking. In *Proc. IUI'06*, 112–123.