Temporal Summaries: Supporting Temporal Categorical Searching, Aggregation and Comparison

Taowei David Wang, Catherine Plaisant, Ben Shneiderman, Neil Spring, David Roseman, Greg Marchand, Vikramjit Mukherjee, and Mark Smith

Abstract—When analyzing thousands of event histories, analysts often want to see the events as an aggregate to detect insights and generate new hypotheses about the data. An analysis tool must emphasize both the *prevalence* and the *temporal ordering* of these events. Additionally, the analysis tool must also support flexible comparisons to allow analysts to gather visual evidence. In a previsous work, we introduced align, rank, and filter (ARF) to accentuate *temporal ordering*. In this paper, we present temporal summaries, an interactive visualization technique that highlights the *prevalence* of event occurrences. Temporal summaries dynamically aggregate events in multiple granularities (year, month, week, day, hour, etc.) for the purpose of spotting trends over time and comparing several groups of records. They provide affordances for analysts to perform temporal range filters. We demonstrate the applicability of this approach in two extensive case studies with analysts who applied temporal summaries to search, filter, and look for patterns in electronic health records and academic records.

Index Terms—Information Visualization, Interaction design, Human-computer interaction, temporal categorical data visualization.

1 Introduction

Generating new hypotheses is an explorative and iterative process. It requires analysts to make generalizations about a dataset. However, making generalizations is particularly problematic when dealing with large numbers of personal records of event data, where each record holds many, and some of which are of the same type. Often analysts must generalize both how events occur in relationship to others events (*temporal ordering*) and also the frequency of the event occurrences (*prevalence*). In doing so, analysts often compare datasets that differ in a single aspect (such as control vs. experimental) in attempt to gather support for a hypothesis. Unfortunately, current analysis tools for temporal event data fall short in accentuating event *temporal ordering* and *prevalence*. They also fail at providing mechanisms for flexible and rapid comparisons.

While command-line query languages provide the means to access large amount of temporal event data, temporal SQL queries are very difficult to specify, and the traditional tabular display of results neither highlights the *temporal ordering* nor exposes their *prevalence*. Our previous work sought to address the temporal ordering problem by using the Align-Rank-Filter (ARF) framework to manipulate the visualization of temporal event data. In particular, we showed in a controlled experiment that using alignment improves user performance greatly on recognizing temporal characteristics among events [21].

While alignment has been well-received by users, our ongoing collaboration with domain experts revealed the following needs in their daily work. They need to be able to specify temporal range constraints in their searches (e.g. find all patients who have had an open-heart surgery *within 3 months* of their first heart attack). They need to view multiple records as an aggregate to study *prevalence* of event data over time. They need to divide and subdivide a set of records into logical groups and subsequently *compare* these groups.

In this paper, we describe our solution – temporal summaries – and how they support these needs. Temporal summaries are stacked bar charts over a time frame. By default, events are the objects of aggregation, and each event type is depicted as a color-coded stack. A single temporal summary allows analysts to compare the distributional trends of multiple event types over time. Applying multiple temporal summaries on multiple groups of records allows analysts to compare these groups of records in a coordinated manner. As analysts zoom in and out, temporal summaries automatically rescale to display the aggregation in the corresponding granularity. Finally, temporal summaries provide affordances for analysts to specify temporal range constraints. This enhances the existing ARF framework, and enables more ways of creating logical sets of groups.

We first present related work. We discuss our design and show the features of temporal summaries through a medical use case. We then present two case studies with our collaborators – the first in medical domain and the second in academia. Finally, we also present the lessons learned from our iterative design process and discuss future work.

2 Related Work

There has been increasing amount of attention to focus on interactive visualization techniques on temporal categorical data. From both the academia and the industry, a number of visual analysis tools have been proposed in a variety of different domains: business intelligence [19], health care and medicine [3][12][15][21], and web session logs [7]. Despite the similarity in goal, there are significant differences, especially on how these tools support aggregation, comparison, and search over temporal data. LifeLines [15] and DataMontage [12] focus on single patient record visualization, and offer very limited search functionalities. An extended version of LifeLines in i2b2 [11] displays multiple patient records at once, but there is no search, aggregation, or comparison features once the data is visualized.

On the other hand, Session Viewer provides these features. It specializes in analyzing logged sessions in search engines [7]. It has a graphical representation of the states sessions are in, and uses thick state transitions lines to show frequent occurrences of events, akin to the visualization approach in [9]. It also uses a stacked bar chart to show event occurrences over time. However, it lacks richer filter mechanisms such as temporal range filters. Session Viewer supports comparison of different session groups, but the support is limited. Analysts cannot dynamically create session groups or compare

- *Taowei David Wang, Catherine Plaisant, Ben Shneiderman are with Human-Computer Interaction Lab and Dept. of Computer Science, University of Maryland at College Park, E-Mail: {tw7, plaisant, ben}@cs.umd.edu.*
- *Neil Spring is with Dept. of Computer Science, University of Maryland at College Park, E-Mail:nspring@cs.umd.edu*
- *David Roseman, Greg Marchand, Vikramjit Mukherjee, and Mark Smith are with ER One Institute, Washington Hospital Center, Medstar Health., E-Mail: {David.H.Roseman, E.G.Marchand,Vikramjit.Mukherjee, mark.s.smith}@medstar.net.*

Manuscript received 31 March 2009; accepted 27 July 2009; posted online 11 October 2009; mailed on 5 October 2009.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org .

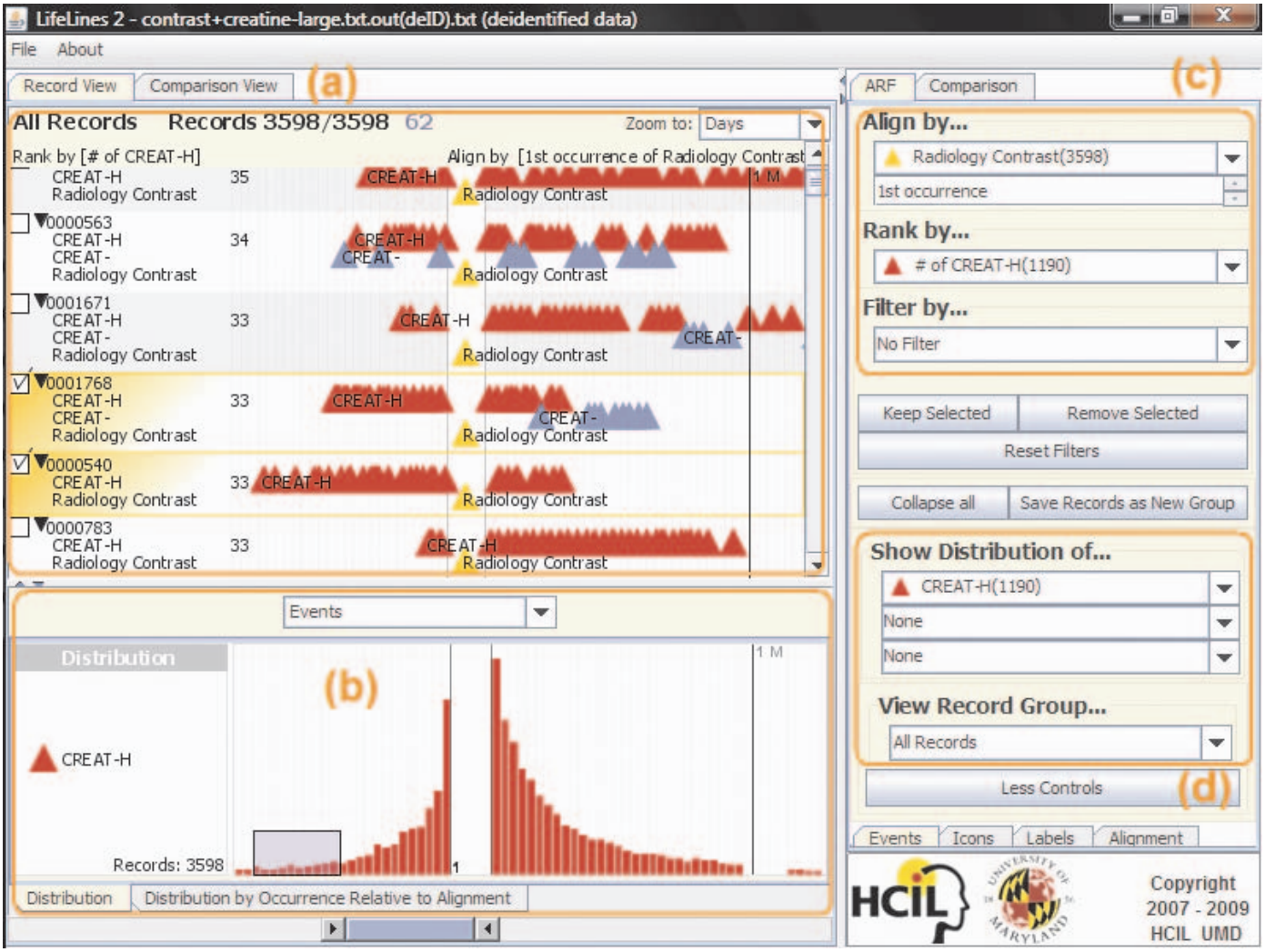

Fig. 1. Annotated Lifelines2 screen shot. (a) Each row represents a record, showing its ID on the left. Each record contains several types of events, listed below its ID. Each type is color-coded, and each instance of event is represented by a colored triangle on the time line. The white vertical band represents the alignment line (aligned by *Radiology Contrast* in orange triangles). A thick black line with the label *1 M* to the right represents one month after the alignment. The combo box to the top right indicates that each finer tick represents a day. The two records in orange gradient (*001768* and *0000540*) indicate that they are currently selected. A temporal summary is shown in (b). The label on the left indicates that it is showing the distribution of the event type *CREAT-H*, which is in red. Each red bar indicates how many events of that type are there in that day. The combo box on top of (b) indicates that we are aggregating events. By changing its value, we can dynamically aggregate by other metrics (e.g. records, or events per record). The transparent blue box in (b) show that the user has drawn a box there to select all records that have at least one *CREAT-H* event in the time the box indicates. The selected records are highlighted in orange gradient in (a), and the total number of selected records are in blue on the top panel of (a). Below (b) is a range slider that can be used to pan and zoom both (a) and (b) in coordination. To the right is the control panel. (c) shows the controls and the current state of Align, Rank, and Filter. (d) shows additional controls for temporal summary, and also for navigating groups. Group creation and other controls are wedged between (c) and (d).

within groups, making it more cumbersome to generate/refine hypotheses. Our approach to comparisons is a more generalized one.

Using concentric circles, Event Tunnel offers a "perspective view" on time, where outside circles represent recent time, and inner circles represent distant time [19]. Analysts can search and filter via type and attribute-based queries, but there is no support for sequence-based or temporal range-based search. Event tunnel offers automatic clustering and allows an analyst to visually compare two different clusters. Though automatic clustering is useful, analysts cannot assert finer control. Like Session Viewer, the visual aggregation does not support investigation over varying temporal granularities.

In contrast to tools with limited search features, PatternFinder [3] and the PatternFinder extension to Amalga [14] allow analysts to build very rich queries for sequences of events in a form-based UI. Although much more expressive than the other interfaces in terms of search capabilities, the complexity to navigate the UI components to specify queries with temporal constraints tend to overwhelm analysts. They also lack aggregation features crucial to overview comparisons.

Aggregating temporal categorical data over time is a common technique for many other applications, but these work offer limited interaction capabilities with the aggregation and even less search capabilities. Dubinko et al. proposed an algorithm to rapidly aggregate photo tagged in different temporal granularities to create compelling photo slide shows [2]. Ribler et al. proposed several aggregation techniques for temporal categorical data, but mostly only focusing on temporal periodicity [17]. ThemeRiver [5] displays keywords from news articles in stylized stacked bar charts to allow analysts detect thematic changes over time. Phan at al. use bar charts to aggregate network event data to analyze network intrusions [13]. Their system allows analysts to use brushing and explicit queries to modify the visualization. Similarly, [22] and [8] use interactive stacked bar charts for analysts to search and spot temporal trends.

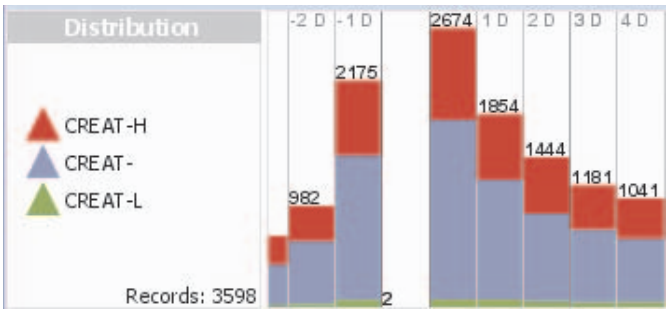

Fig. 2. A temporal summary showing the daily distribution of creatinine test results of 3598 patients when aligned by their 1st occurrence of *Radiology Contrast*.

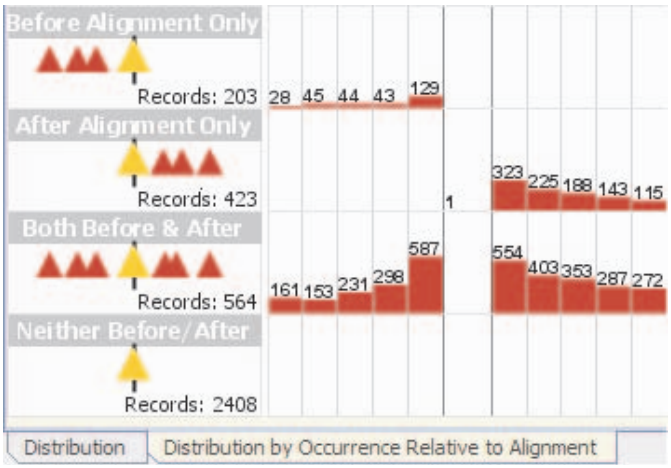

Fig. 3. The second tab in (b) of Figure 1 is activated. The data in Figure 1 is split into 4 mutually exclusive groups: those who have *CREAT-H* only before alignment, after, both, or neither. 2408 patients never had any *CREAT-H*. 564 patients have *CREAT-H* both before and after the alignment.

However, these previous work lack the more advanced analysis features, such as temporal range filter, our approach supports.

Although numerical time series visualizations fundamentally support very different tasks, many existing work present multiple time series in coordination, similar to how we present comparison of temporal summaries. Among others, [4] and [16] present multiple numerical medical data of a single patient in the same view to facilitate visual tracking of a patient's condition, but offer none or very limited interactions. Interactive Parallel Bar Charts [1] and TimeSearcher [6] visualize multiple numerical time series data for coordinated exploration. They both offer direct selection over a temporal range as a basis for a pattern search. Our approach also includes a direct-manipulation visual query method with the temporal summaries, but our focus is on event occurrences within temporal constraints, not on numeric patterns. We know the difficulties analysts have with large form-based UI to specify temporal constraints [3], so we mimicked the successful TimeBox from [6]. We let analysts draw one or more blue boxes in temporal summaries restrict time frames and filter records (Figure 1 (b)).

3 Temporal Summaries

Our previous work Lifelines2 brings *temporal ordering* of event data to the analysts' attention via aligning sentinel events [21]. However, even with alignment, analysts are still burdened to visually scan the entire dataset to make sense of and to make generalizations about the data. To overcome this shortcoming, we added temporal summaries and its appropriate controls ((b) and (d) in Figure 1). A temporal summary is a stacked bar chart that aggregates a variety of metrics (event counts, record counts, etc.) about a group of records over a time frame of varying granularities.

Consider the medical scenario: some patients who undergo a radiology contrast experience adverse reaction and as a result, their renal functions decline. If not noticed and taken care of, the condition may be fatal. To monitor a patient's renal function, blood tests are performed regularly both before and after the radiology contrast to monitor the creatinine level. High creatinine indicates lowered renal function. The adverse reaction usually occurs within 14 days after a radiology contrast. We use this scenario as a concrete example to introduce the features of temporal summaries. We show how analysts utilize temporal summaries to see patterns, generate new hypotheses, and possibly change the direction of visual exploration. More specifically, we show how temporal summaries work with the existing ARF framework to find the patients who may have this adverse reaction. We also show how analysts dynamically create subgroups of records, and use temporal summaries to perform comparisons among them.

3.1 The Existing Align, Rank and Filter Framework

The existing Align, Rank, and Filter framework allows analysts to visually align all records by a chosen n^{th} event (whether it is from start of the record or from the end), or by all occurrences of that event. When "all occurrences" is chosen, Lifelines2 duplicates records that have more than 1 such event. Analysts can rank records by number of occurrences of an event type. For example, Figure 1 shows that all records are aligned by the 1st occurrence *Radiollogy Contrast* and ranked by number of occurrences of *CREAT-H* (creatinine high) so that the records that have the most are displayed on top. The number next to each record's header on the left indicates how many *CREAT-H* events that record has. Lifelines2 provides several filter mechanisms. First, analysts can select an event type and a number to filter by number of occurrences of an event. Analysts can also filter by a sequence. By selecting from a list of drop down boxes, analysts can specify a temporal sequence, and remove records that do not contain such sequence. A functional improvement over our previous work [21] is that the analysts can now specify both event presences and event absences in the sequence filter. That is, analysts can specify A before C, with no B in between. The sequence filter does not afford ways to specify temporal constraints such as B after A *within 2 days*.

In (b) of Figure 1, the temporal summary is aggregating the event *CREAT-H* (creatinine-high) over 3598 records and over the visible time frame. Analysts can select up to three types of events from the controls in (d). When multiple events are selected, they stack up, and analysts can visually compare the relative proportions over time (Figure 2). The combo box on the top right of (a) indicates that the aggregation is in the granularity of days. Temporal summaries will automatically aggregate over an hour or over a month as analysts zoom in or out. The combo box on top of (b) indicates that the summary is aggregating events, but analysts can make it aggregate records instead (how many records have at least one event of the specified types in each day) or events per record by controlling the combo box. Analysts can directly draw one or more selection boxes onto temporal summaries to specify a temporal range selection (Figure 1 (b)). Records that are selected are highlighted in orange gradient with check marks next to them (Figure 1 (a)). By affording a way to specify multiple temporal range filters, temporal summaries enhance ARF.

3.2 Comparing within a Single Group

3.2.1 Showing Distributions of Multiple Events

Medical analysts would bring up Liflines2 and rank by *CREAT-H* events to bring the most "severe" patients up top and align by the 1st *Radiology Contrast*. Then they would bring up temporal summary and show the distribution of *CREAT-H* as in Figure 1. The medical analyst may hypothesize that either the patients had worse renal function immediately before and after their first radiology contrast, or that more tests had been performed closer to a *Radiology Contrast*. Adding all other creatinine test results *CREAT-* (normal),

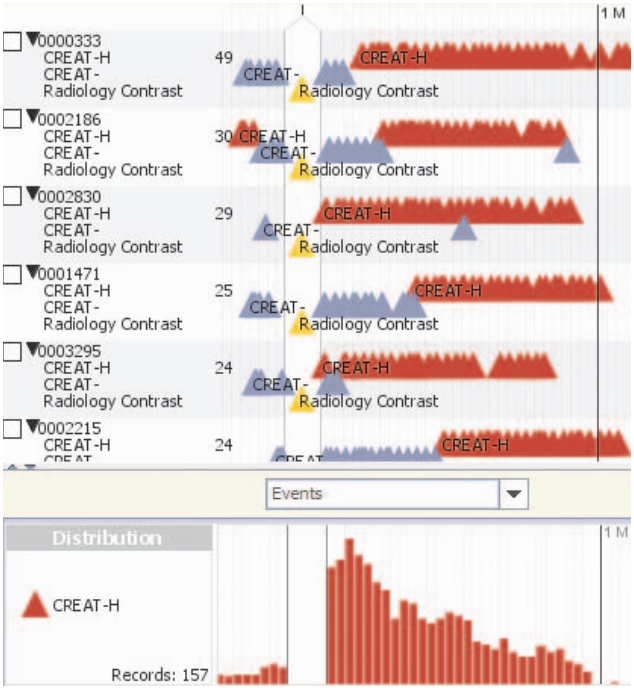

Fig. 4. The first 6 patients in the final group of 157 that fits our filtering criteria. The unusual peak of high creatinine tests on the third day after the alignment is indicative that our exploratory search is heading to the right direction.

CREAT-L (low) to the temporal summary indicated that it is more likely the second case, as more tests were performed near the alignment (Figure 2).

3.2.2 Splitting Record by Event Occurrences

The second tab at the bottom of the temporal summary in Figure 1 lets analysts split the records in view into four mutually exclusive groups, and display the 4 summaries simultaneously. These records are split by the occurrences of their *CREAT-H* events with respect to the alignment. The records that have the *CREAT-H* events only before the alignment are in the first group, the records that had *CREAT-H* only after are in the second group. Those that had *CREAT-H* both before and after were in the third group, and those that had *CREAT-H* in neither were in the fourth group. Events that co-occur at the same moment as the alignment event are not considered in deciding which group they fall into, unless it is the only such event in the record (in which case, the record will be classified into the last group). Figure 3 shows the respective temporal summaries. The third summary shows that over 500 patients have *CREAT-H* before the alignment. Our medical collaborators believed that many patients in that subgroup probably experienced chronic renal deficiency, and should be removed from our data as their lowered renal function was not related to the radiology contrasts. Upon our physicians' suggestion, we used the occurrence filter to remove those who have had at least 4 *CREAT-H* prior to the alignment (using the "Remove Selected" and "Save Records As New Group" buttons on the control panel on the right). This removed 219 records.

3.2.3 Direct Manipulation Temporal Range Filters

Next we aligned the rest of the records by all occurrences of *Radiology Contrast*. Lifelines2 duplicated the records that have multiple *Radiology Contrasts*, and placed the duplicates and the original in accordance with the alignment in the same display. After applying the alignment, the temporal summary aggregated over all of the duplicates and originals in the temporal summary. This means when we specified a selection box over the summary now for the first 2 weeks after the alignment, we were selecting all patients who had at least one *CREAT-H* events within 2 weeks after any *Radiology Contrasts*. Multiple-event alignment like this one allows us to select all occurrences with respect to a single event, and the temporal summary allows us to specify the temporal constraints. Once a selection is made (either through selection from temporal summaries, or through the filtering mechanisms), analysts can create a new group of records and give it a name that they can refer back to, using the controls nestled between (c) and (d) in Figure 1.

This phase of filtering reduced the patient count to 792, but when our collaborators looked at them, almost 90% of the patients in this group did not display the temporal characteristics of contrast-related renal deficiency. Our collaborators recommended to restrict only patients who have a baseline reading of normal creatinine reading prior to contrast. We applied a sequence filter that specified patients who had a normal reading of creatinine, followed by *no* readings of high creatinine, followed by a radiology contrast, and finally followed by a reading of high creatinine. Finally, this resulted in just 157 patients (Figure 4).

Since we ranked all patients by the number of *CREAT-H* events in descending order, the first six patients in figure 4 are likely to be the most severe. Aside from severity, these patients also showed strong evidence for contrast-related renal deficiency in the two-week timeframe after the alignment. The temporal summary showed a peak of *CREAT-H* event on the third day after the alignment, an unusual pattern. Summaries of all previously created groups (not shown) showed a steady decline of the *CREAT-H* counts after *Radiology Contrasts*. This anomaly suggests that our exploration was in the right direction. After removing patients who had 2 *CREAT-H* or fewer events as unlikely candidates, we presented our collaborators a total of 84 potential patients who experienced contrast-related renal deficiency.

3.3 Comparing Between Groups

The previous sections show that temporal summaries are useful showing event distributions of a single group over time. They are data overviews that help analysts make decisions on how to proceed with an exploration. They are also tightly integrated with the existing ARF framework to enhance searching. But temporal summaries are most useful when used to compare multiple groups of records at once. Analysts can select "Comparison View" (a tab on top of (a) in Figure 1) to compare automatically created groups (Section 3.2.2), or compare previously created groups. In Comparison View, analysts can specify any number of events (instead of only three) to be included in the comparison, and align the data just as before. If analysts choose to perform a within-group comparison, they must also specify a split criterion. We have already seen split by event occurrences relative to the alignment in Section 3.2.2. Analysts can also split by whether a record has N numbers of a certain event, automatically creating 2 mutually exclusive groups. When between-group comparison is selected, analysts can select multiple previously created groups and explore these groups using temporal summaries in a coordinated manner. It is worth noting that the automatically split groups are always mutually exclusive, while analyst-created groups are not constrained by that.

Figure 5 shows a partial screen shot of the Comparison View. The controls to the right indicate that this is a between-group comparison. The final group *Final 157* and its complement are selected and shown as two summaries. The creatinine high (*CREAT-H*) and normal (*CREAT-*) events are aggregated by month. All patients are aligned by their first radiology contrast. When comparing groups that have a large difference in the number of records (i.e. 3441 vs. 157), a raw count of events is often not informative. The counts need to be normalized by the number of records in each group in order for meaningful comparison. Using the "Events (Normalized by Records)" display option, as in Figure 5, the event counts are normalized. The numbers on top of the bars indicate how many creatinine high and normal readings per patient in that month. This comparison shows that the *Final 157* patients had a

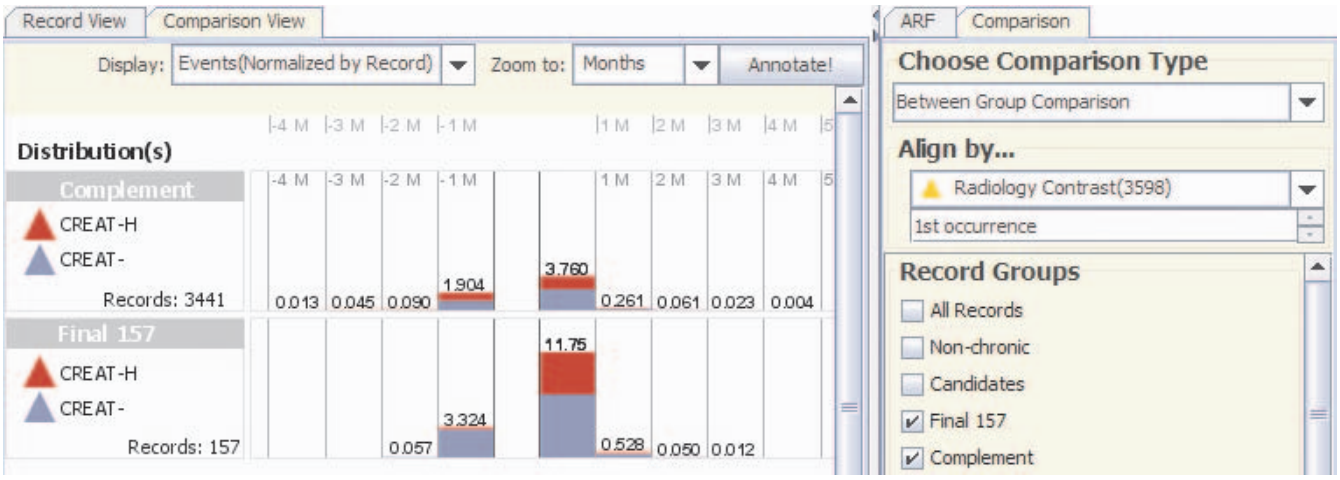

Fig. 5. Two summaries are shown in this between-group comparison. One is the result of the final filter (*Final 157*), and the other is the complement (*Complement*) set. Creatinine high (*CREAT-H*) and creatinine normal (*CREAT-*) are aggregated by month. The events counts here are normalized by the number of records in that group for meaningful comparison. The numbers on top of the bars indicate the average number of *CREAT-H* and *CREAT-* events per record in that month. The patients in *Final 157* clearly have much higher normalized averages.

much higher average than its complement. This is because the patients in the *Final 157* are higher-risk patients, and tests were performed more frequently for each patient to better monitor their health.

The contrast-creatinine scenario was developed with our physicians over a period of six months. We iteratively refined our design and added features to support their needs. The successful process of narrowing the set of patients and showing comparisons pleased our analysts. We were able to build confidence in the value of Lifelines2, and subsequently proceeded with our case studies.

4 Case Studies

We present two case studies here. The first is another application in medical records where we are studying the length of stay of patients in the hospital. We also report a second case study on graduate student academic records. Both of these studies focus on searching for specific groups of records, and comparing among different groups to evaluate the analysts' hypotheses, although the medical scenario is more mature.

4.1 Heparin-Induced Thrombocytopenia

Thrombocytopenia is a medical condition in which the platelet count in blood stream is low. Heparin, a drug used as an anticoagulant, is known to cause this adverse side effect in 0.5-10% of patients. Heparin-induced thrombocytopenia (HIT) is characterized by a sharp (usually greater than 50%) drop of platelet counts within 5 to 9 days after the first administration of heparin. However, not all drops indicate HIT. Lowered platelet count can simply be the normal side effects of heparin. To ascertain whether a patient has HIT, an additional test called HIT antibody test is ordered. Unfortunately, the HIT antibody test has high sensitivity but low specificity. When the test returns negative, then the patient is likely not to have HIT (98% accuracy). But when the result is positive, the test is only about 25% accurate. In clinical care, a hospital does not have the luxury to spend an additional 5-7 days to perform a more accurate test. Instead, patients whose HIT antibody test returns true are treated as if they have HIT. A recent medical study on 22 patients showed that a hospital treating 50 HIT patients a year can incur $70,000 to $1,000,000 and each patient can increase length of hospital stay by at least 14.5 days [18]. These patients can increase the financial cost and stretch resources of a healthcare facility.

Our physician partners at Washington Hospital Center are interested in verifying these results and see if clinical care data differ from the study. In particular, they want to focus on the patients who have been admitted to the intensive care units (ICU). Our collaborators could query for the relevant data in their medical database and perform the analysis that way with the help of the database administrator. However, they would rather see the temporal ordering of these events and interactively narrow the data down because in order to determine whether a patient actually has HIT, the temporal ordering of events and the temporal constraints are important.

Over a period of one and half months, the developer (Wang) and an additional case study observer (Plaisant) visited Washington Hospital Center three times to meet with the physician collaborators (Mukherjee, Smith) and the database administrator (Roseman). Each meeting lasted approximately two hours. Much work had been devoted to understand the medical database and to clean it up to make it suitable for this case study. Over this period of time, Roseman and Wang worked via email to obtain de-identified medical data, and converted them into a format Lifelines2 accepts. Microsoft Amalga [10] was the clinical information system that served as the data source for the de-identified data on which the heparin-induced thrombocytopenia case study was conducted. Its data-centric architecture enabled relatively easy extraction of the requisite dataset. Over all, there were over 30 emails exchanged between the developer and the collaborators to discuss the topic, to refine or to add data, and to decide on the logistics (when and where to meet, etc.). When all of us met face-to-face, we spent most of the time exploring the data using Lifelines2 together. In the following exposition, “we” is used to include everyone involved in the case study.

We obtained de-identified data on all 841 patients who visited the hospital and had a HIT test for the calendar year of 2008. For each patient, we have the medical designation of platelet counts in categories (High, Normal, Low, Critical), HIT test results (Positive, Negative, Borderline), administration of any of the 9 heparin variants, admission and release from ICUs, and discharge code from the hospital (Dead or Alive). The categories were further pre-processed to include higher level categories. For example, platelet Normal and High events are considered the same in this investigation, so we created the new category High/Normal (while also keeping the existing High and Normal categories just in case) to facilitate our exploration.

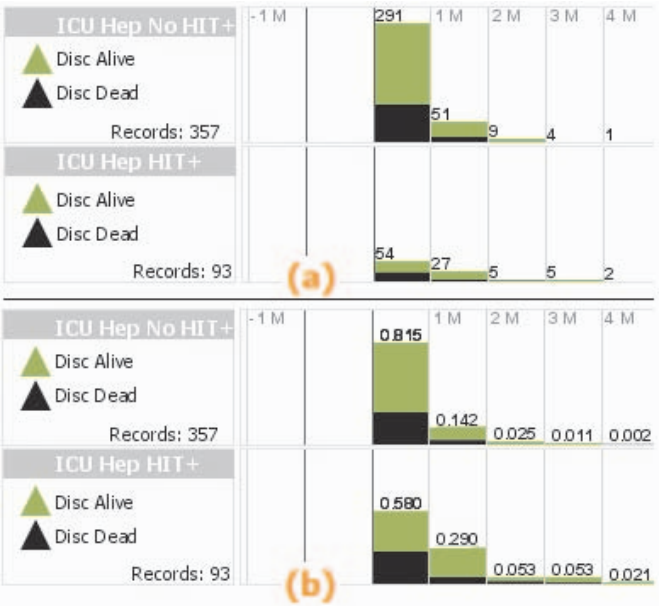

Fig. 6. The temporal summaries show discharge patterns (*Discharged Alive* in green, *Discharged Dead* in black) aligned by the first admission to ICU. In (a), the raw count of event are shown, but the large disparity in number of patients between the two groups makes it hard to compare. In (b) the counts are normalized by the number of patients. It is clear to see that patients in *ICU Hep HIT+* tended to stay longer in the hospital than those in *ICU Hep No HIT+*, where over 80% of patients in were discharged within 1 month.

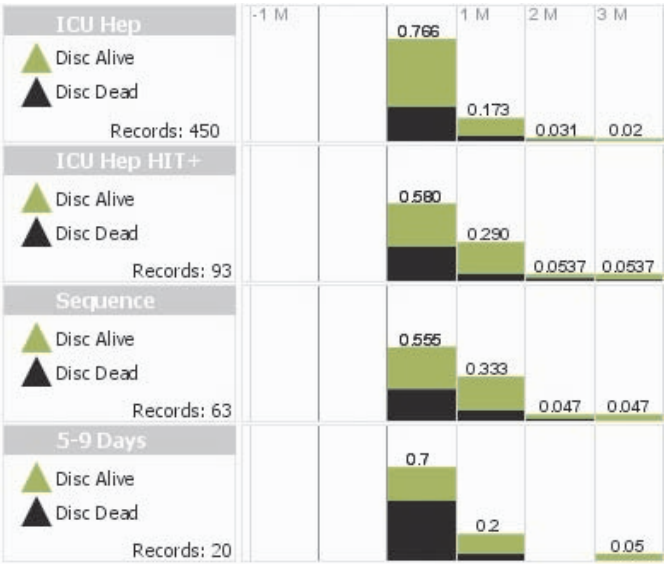

Fig. 7. Normalized hospital discharge data aligned by first admission to ICU from 4 groups are compared here *Discharged Alive* in green, *Discharged Dead* in black). Each group is a subset of the one above it, and a "closer" approximation to true HIT patients. We hypothesized that the closer approximation, the more stretched the discharge pattern should be. The first three seem to support our hypothesis, but not the last one -- we see that there are far more *Discharged Dead* than the others in the first month, and this may be skewing the data.

From the original dataset, we filtered to find patients who were admitted to the ICU and also had exposure to heparin. This *ICU-HEP* group has 450 patients. Then we applied another filter, to divide this new group into two subgroups: the 93 patients who had a HIT positive test (*ICU-HEP-HIT+*), and the 357 who had not (*ICU-HEP-NoHIT+*). The hypothesis is that there is a difference in the length of hospital stay between those who may have HIT, and those who almost certainly do not have HIT. We aligned the patients by their first admission to ICU, and compared these two groups against each other to see if there are noticeable differences in the distribution of discharge events (Figure 6 (a)).

The large difference in patient numbers (93 vs. 357) makes comparison of raw counts meaningless and also impossible for our physician partners to detect trends. We normalized the counts by selecting "Events (Normalized by Records)". In the new summaries, the counts are normalized by the number of patients in each group, and the bar heights are normalized across the two summaries for direct visual comparison (Figure 6 (b)). It was easy for our collaborators to recognize that the discharge distribution in *ICU-Hep-HIT+* looked more stretched out (wider and shorter), indicating that the patients tended to stay in the hospital longer when they had a positive HIT antibody test result.

We further created more subgroups from *ICU-Hep-HIT+*, ones that approximated the ideal temporal orderings of HIT patients closer. Our physician partners hypothesized that by narrowing down to patients who are more likely to actually have HIT, the discharge patterns in temporal summaries may stretch even more. First, we used the sequence filter to find those who had never had a *Platelet Low/Critical*, followed by a *Platelet Normal/High*, followed by any type of *Heparin*, followed by *Platelet Low/Critical*, and finally followed by a *HIT Positive* test. The filter identifies only patients who had only normal levels of platelets up until they were exposed to heparin, after which they experienced a drop in platelet, and a HIT positive result was returned. This group is named *Sequence*, and contains 63 patients. From *Sequence*, we then selected, via temporal range filter in the temporal summaries, only those who had *HIT Positive* results within 5-9 days after their first exposure to *Heparin*. This *5-9 Day* group has 20 patients.

The hypothesis is that as we used more stringent filters to create patient groups that better approximate the true group of patients who have HIT, we expected see the discharge pattern to be more and more spread out. The comparison of these 4 groups in Figure 7 showed that while that seems to be the general trend in the first three groups, the last *5-9 Days* group do not follow this trend. We believe it is due to the small number of patients in that group and the higher-than-average number of *Discharged Dead* patients in the first month.

Through these exploratory analysis exercises, we have found that for patients in ICU, those who had *HIT Positive* tended to stay in the hospital longer than those who had not. Extended stay in the ICU generally translates to increased cost, but we thought it might be worth it to explore with just the data we have. We wanted to see if the patients who are approximate better to true HIT patients received more resources in terms of number of platelet tests performed. In this comparison, we included the last three groups in Figure 7 and also the group of patients who were admitted to ICU, had exposure to heparin, and had a *negative* result in the HIT test. We aligned by each patient's first admission to ICU and compared the normalized platelet data to see how many platelet tests were performed per patient in each month (Figure 8). We had expected to see the lower groups to have higher number of platelet tests, but there was little visual evidence to support that hypothesis. There was indeed a large difference in the number of platelet tests per patient in each month between the first and the second group, but there was little difference among the other three. The reason is that the hospital treats all HIT test positive patients (the last three groups) with the same diligence and caution even though the HIT test is only 25% accurate when it is positive.

We were pleased that Lifelines2 succeeded in allowing our physician partners investigate and gather visual evidence with respect to their hypotheses. The comparisons on the discharge pattern showed that the data seems to support the hypothesis that HIT patients tended to stay in the hospital longer. However, because hospitals do not know whether a patient has HIT *a priori* and can only rely on the result of the low-sensitivity HIT test, we see the

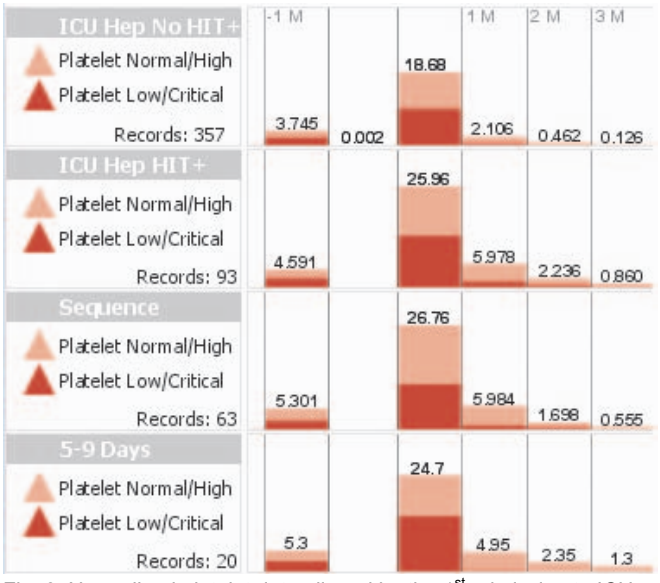

Fig. 8. Normalized platelet data aligned by the 1st admission to ICU for 4 groups (*Platelet Normal/High* in pink, and *Platelet Low/Critical* in red). We see these numbers of platelet tests only dramatically increase from the first group to the second.

evidence that the hospital treats all of the HIT positive patients with heightened diligence and care with regards to monitoring platelets. If it were true that HIT patients do incur more cost, the cost would have to come from elsewhere. It is interesting to note that although both are comparisons of categorical data, the first comparison highlights the behaviour of patients (how soon they get well with the hospital's help), while the second highlights the behaviour of the hospital (how well the hospital treats the patients).

4.2 Monitoring Graduate Student Progress

A second application of the temporal summaries is to monitor and evaluate graduate student progress. Progress through a PhD program can be measured loosely by grades in course work, advancing through program milestones, publishing research papers, etc. Each year, the faculty of our department review the progress of each student, considering each of these factors, with the intent to offer advice to students and their advisors. The tool described in this paper is the first visualization tool to be applied to the process.

With various queries in SQL, the review can identify students who have fallen behind or are approaching a program deadline. However, temporal summaries can help to solve two classes of question that SQL supports poorly. First, are there factors that may predict falling behind schedule? Below, we ask how well being a teaching assistant (TA) for four semesters or more predicts a longer time to advance to candidacy. Second, is there evidence that the graduate review helps students be better aware of milestones and make better progress? The review has incidental benefit through, for example, making all advisors aware of graduate program deadlines, but quantitative evaluation is difficult.

There are three fundamental differences between student time lines and patient histories in the other studies. First, although the "outcome" in the medical setting is clear (months spent in the hospital, how patients were discharged), the outcome for a graduate student is much less well defined. Further, we consider and maintain only the information that describes currently-enrolled students; those who have left the program without a degree, who are on-leave, or even have completed the program, are not evaluated and are not (currently) in the graduate review data. This limitation in the data reduces the precision of any conclusions: for example, of the population of students who entered the program six years ago, only those who have not yet completed their dissertations are included, potentially increasing time-to-milestone statistics. Second, the confidentiality of current student records and the lack of a good data de-identifier constrain how we present results: We do not include screen-shots for this reason. Third, student time lines do not precisely match chronology: students may start in the spring or take a leave of absence, adjusting how time spent in the program corresponds to real time.

How might we use detailed information about a student's time-line to better predict timely completion of milestones? After an introduction to the tool, the analyst set about to determine how spending many semesters as a TA affected the time to propose a thesis. Events represented the start of a graduate career, each semester as a TA, and advancing to candidacy. To validate the hypothesis that more TA'ing implied a longer time to graduation, the analyst constructed three groups: those who had advanced, those who had advanced after four or more semesters of TA'ing, and those who had advanced after three or fewer. To do so he aligned all students by the *Advanced* to candidacy event, implicitly selecting only those students who had advanced. From this group, he used filters to choose two disjoint, approximately equally sized sub-populations: those who had TA'ed four or more semesters, and those who had TA'ed three or fewer semesters. When comparing these two groups and the union, it appears that, indeed, it tends to take students who TA four or more semesters on average one additional year to propose a thesis. Whether time spent TA'ing is a cause of delay or merely a symptom is not easily determined, but that does not diminish its predictive value.

To quickly evaluate the potential benefit of the graduate review, the analyst next constructed groups of students who were classified (by the SQL-query-based tests) as falling behind schedule to explore their success in later years. At each annual review, each student is given a high-level categorization that attempts to capture whether the student is *On Target*, *Concerned*, or *Very Concerned*. After aligning students by the first occurrence of *Concerned*, the events that followed showed that over 70% of the students who received such a mark were no longer marked as under *Concerned* the following year.

These analyses are preliminary; their results are not demonstrably true because of the biases in the partial dataset. However, to assist the analysts in exploring the data to quickly test simple hypotheses, the temporal summary provided significant help in answering key questions about the review of student progress. This initial portion of a continuing user study was conducted over a month. The developer and the analyst met and worked together for about four hours. The analyst drove the application and spent significant amount of time using the tool on his own. However, the developer and the analyst communicated steadily via e-mails.

5 User Impressions and Discussions

The two case studies were conducted differently. In the HIT study, all the exploration was done collaboratively, but our medical collaborators dictated what operations to apply, and the developer "drove" the software. In the student record case study, all actions were performed by our collaborator, and he continued using the tool independently. Although the HIT study was not driven by the analysts, more aspects of the application were used in more depth.

Our medical collaborators were happy to see patient records on a time line and not in a spread sheet. One said, "I am a very visual person, and the events laid out this way corresponds exactly to how I think." They did not seem to have problems understanding ARF or interpreting the data the first time they saw it. After a 10-minute introductory demonstration of the features, our collaborators were in control, dictating what steps to take in the exploration. The dictations included both the high level logical and also what specific operations to take, reflecting their full grasp of the features.

On the other hand, having to drive the application through someone else's dictation revealed a lot of room for improvements. The exploratory steps in this case study involved a lot of group creation, between-group and between-view navigation. When creating a large number of groups, better management is needed. Additional group creation mechanisms such as taking the

intersection of two chosen groups would provide for quicker results. Group names alone are not enough to help analysts keep track of the groups. Provenance information such as "what filtering mechanisms did I use to create this group" needs to be automatically saved to help analysts remember. We had designed the ARF framework so each group retains its own current state of align, rank, and filter. However, this case study revealed that analysts would want to apply the same ARF to all groups, and track how these groups differ. Allowing linked exploration among all groups would be helpful.

In the student record scenario, our analyst was driving the application, so the developer could observe how analysts unfamiliar with Lifelines2 might use it. We observed that both ARF and the temporal summaries were easily grasped. The analyst commented, "Alignment is so useful". However, the nuances on how to better strategically use ARF escaped our analyst in early use, which confirmed findings from our previous controlled experiment [21]. In a few instances, the analyst asked if it was possible to create a certain group. The process required several steps and a combination of more than one selection mechanisms. A new user who had spent fewer than 10-15 minutes with the application was not expected to make that kind of connection. On the other hand, selecting from temporal summaries and grouping selected records were understood well in that time frame. Our collaborator used temporal summaries to perform temporal selections and create additional groups with relative ease.

6 Conclusions and Future Work

We present temporal summaries, a stacked bar chart that aggregates event data over multiple personal records in varying time granularities. Single temporal summaries allow analysts to study trends of multiple event types. Analysts can also dynamically subdivide a dataset into smaller groups of records and utilize temporal summaries to compare event trends among these groups. Temporal summaries allow analysts to apply temporal range filtering via direct manipulation, sidestepping complex UI widgets that often overwhelm analysts. Finally, we show in two case studies that these interactive and visualization features allow analysts to gather visual evidence, generate new hypotheses, and alter the path of their exploratory process by accentuating *temporal ordering* and *prevalence* of events.

Although the feedback is encouraging, the case studies revealed several usability issues in our system. Group management, provenance information, and lack of linked ARF in all groups are problems that hinder analyses. We were happy to see that it took minimum time get our collaborators to conceptually understand the features. In particular, we were pleased to see the temporal range selection extensively used in temporal summaries. We will continue working with our collaborators in these domains to help them succeed and improve our system.

Additional future work includes providing better metadata support (events associated with other data). In the spirit of John Tukey's proposal [20], "exploratory and confirmatory can – and should – proceed side by side," we also imagine adding statistics to enrich the temporal summary visualization.

Acknowledgments

The authors would like to thank Mike Gillam and Shawn Murphy. This work was supported by a grant from ER One Institute, Washington Hospital Center, Medstar Health.

References

[1] L. Chittaro, C. Combi, and G. Trapasso, "Data Mining on Temporal Data: A Visual Approach and its Clinical Application to Hemodialysis," *Journal of Visual Languages and Computing*, 14, 6, 591-620, 2003.

[2] M. Dubinko, R. Kumar, J. Magnani, J. Novak, P. Raghavan, A. Tomkins, "Visualizing Tags Over Time", ACM Transactions on the Web, 1, 2, 2007.

[3] J. Fails, A. Karlson, L. Shahamat, and B. Shneiderman, "A Visual Interface for Multivariate Temporal Data: Finding Patterns of Events Over Time", *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, 2006.

[4] A. Faiola and S. Hillier, "Multivariate Relational Visualization of Complex Clinical Datasets in a Critical Care Setting: A Data Visualization Interactive Prototype," *Proceedings of the International Conference on Information Visualization*, 460-468, 2006.

[5] S. Havre, E. Hetzler, P. Whitney, and L. Nowell, "ThemeRiver: Visualizing Thematic Changes in Large Document Collections", *IEEE Transactions on Visualization and Computer Graphics*, 8, 1, 9-20, 2002.

[6] H. Hochheiser and B. Shneiderman, "Dynamic Query Tools for Time Series Data Sets, Timebox Widgets for Interactive Exploration," *Information Visualization*, 3, 1, 1-18, 2004.

[7] H. Lam, D. Russell, D. Tang, and T. Munzner, "Session Viewer: Visual Exploratory Analysis of Web Session Logs," *Proceedings of IEEE Symposium on Visual Analyitcs Science and Technology*,147-154, 2007.

[8] J. Lee, "Exploring Global Terrorism Data: A Web-based Visualization of Temporal Data," *ACM Crossroads*, 15, 2, 7-16, 2008.

[9] J. Lin, E. Keogh, S. Lonardi, J.P. Lankford, and D.M. Nystrom, "VizTree: A Tool For Visually Mining and Monitoring Massive Time Series Databases," *Proceedings of the International Conference on Very Large Data Bases '04*, 30, 1269-1272, 2004.

[10] Microsoft Amalga, http://www.microsoft.com/amalga/

[11] S.N. Murphy, M. Mendis, K. Hackett, R. Kuttan, W. Pan, L. Phillips, V. Gainer, D. Berkowicz, J. Glaser, I. Kohane, and H. Chueh, "Architecture of the Open-source Clinical Research Chart from Informatics for Integrating Biology and the Bedside," *Proceedings of the American Medical Information Association Annual Symposium '07*, 548-552, 2007.

[12] J. Ong, DataMontage, http://www.stottlerhenke.com/datamontage, 2006

[13] D. Phan, J. Gerth, M. Lee, A. Paepcke, and T. Winograd, "Visual Analysis of Network Flow Data with Timelines and Event Plots," *Proceedings of the workshop on Visualization for Computer Security 08*, 85-99, 2008.

[14] C. Plaisant, S. Lam, B. Shneiderman, M. Smith, D. Roseman, G. Marchand, M. Gillam, C. Feied, J. Handler, and H. Rappaport,"Searching Electronic Health Records for Temporal Patterns in Patient Histories: A Case Study with Microsoft Amalga," *Proceedings of the American Medical Information Association Annual Fall Symposium '08*, 2008.

[15] C. Plaisant, R. Mushlin, A. Snyder, J. Li, D. Heller, and B. Shneiderman, "LifeLines: Using Visualization to Enhance Navigation and Analysis of Patient Records", Proceedings of the American Medical Information Association Annual Fall Symposium '98, 76-80, 1998.

[16] S.M. Powsner and E.R. Tufte, "Graphical Summary of Patient Status," The Lancet, 344, 8919, 386-389.

[17] R. Ribler, A. Mathur, and M. Abrams, "Visualizing and Modeling Categorical Time Series Data," *In Symposium on Visualizing Time-Varying Data*, 3-19, 1995.

[18] M. Smythe, J.M. Koerber, and M. Fitzgerald, J.C. Mattson, "Financial Impact of Heparin-Induced Thrombocytopenia," *Chest Journal*, 134, 3, 568-573, 2008.

[19] M. Suntinger, J. Schiefer, H. Obweger, and M.E. Groller, "The Event Tunnel: Interactive Visualization of Complex Event Streams for Business process Pattern analysis," *Proceedings of IEEE Pacific Visualization Symposium '08*, 111-118, 2008.

[20] J. Tukey, *Exploratory Data Analysis*, Addison-Wesley, 1977.

[21] T.D. Wang, C. Plaisant, A.J. Quinn, R. Stanchak, S. Murphy, and S. Shneiderman, "Aligning Temporal Data by Sentinel Events: Discovering Patterns in Electronic Health Records," *Proceedings of SIGCHI Conference on Human Factors in Computing Systems*, 457-466, 2008.

[22] M. Wattenberg, "Baby Names, Visualization, and Social Data Analysis", *Proceedings of IEEE Symposium on Information Visualization* '05, 1-7, 2005.