Patternfinder 3.0: Sparse Temporal Data Visual Query Application

Hyunyoung Song, Nathaniel Ayewah, Gleneesha Johnson

Department of Computer Science,

University of Maryland,

College Park, MD 20742, U.S.A

{ hsong, ayewah, gjohnson}@cs.umd.edu

Abstract

Sparse temporal datasets are often stored and queried using traditional relational schemas and query languages. But this approach does not give users an easy way to search for data points using the temporal relationships between them.

Patternfinder is a form-based visual query interface for querying sparse temporal datasets. It simplifies the query formulation process and visualizes the results using an overview display that reveals temporal patterns. Using Patternfinder, clinicians or physicians can search for undesirable patterns in a patient's medical history.

1 INTRODUCTION

In sparse temporal datasets it is important to have a querying system to detect patterns. This general task raises questions about how to model the dataset, how to query the dataset and how to visualize the results.

In high intensity medical environments such as Emergency Departments (ED), clinicians should become productive immediately upon arrival. Users in this environment often prefer a visual interface over text based systems. Less intuitive systems require more training time and are more likely to lead to user errors.

This paper presents Patternfinder, an interactive interface that builds on previous research in visual and dynamic query systems. Patternfinder has been applied to a temporal data set in the medical domain from Washington National Hospital Center.

The temporal dataset was extracted from a relational database that has time stamps associated with each tuple. Using Patternfinder's visual query interface, doctors or clinicians can create most commonly used queries. The visualization panel will display the result set in a timeline that gives an overview of the result and detailed information on each record.

2 PREVIOUS WORK

There are two paths of research on temporal datasets. One is temporal reasoning, which uses artificial intelligence and data mining for tasks such as diagnosis, monitoring, projection, forecasting and planning [1]. Another path involves providing an interactive interface for dynamic visualization of the temporal dataset. This paper extends several approaches in this second path.

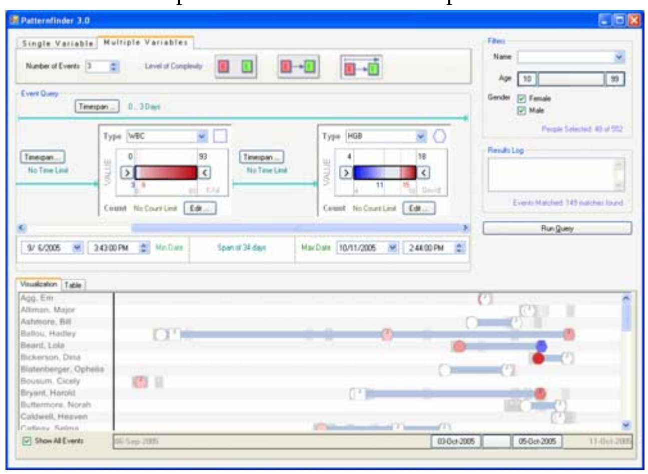


Figure 1 Patternfinder 3.0 is a temporal data query interface with a form-based visual query interface (upper half) and a result visualization panel (lower half).

2.1 Abstraction of Time Oriented Data

Temporal data models are often derived from general relational or object-oriented data models which do not have an internal concept of time. Introducing a time dimension to each tuple is the simplest approach [2]. In ARAMIS system, each tuple is treated as an event and contains patient ID, time and clinical parameter type attributes. The dataset can be indexed by any of these 3 fields as in a relational database. Patternfinder adopts this approach because patient history information is often kept in relational databases that have these fields.

Another approach is to model temporal features (such as intervals or temporal relationships) into time-oriented entities. In HyperLipid system [3] temporal features are modeled into 3 types of entities: events (time points), therapies and phases. Therapies are intervals between two time points and the conceptual location of a phase is in between these two.

The RESUME system [4] combines these two approaches by time stamping the data and using domain specific knowledge to produce interval-based concepts at the same or higher level of abstraction as the original data. This facilitates exploration of the data at multiple levels of abstraction. This system exposes the impact that domain knowledge has on the way temporal datasets are modeled.

2.2 Visual Temporal Query System

Query languages for non-temporal data models (such as SQL) can be used to query temporal datasets. But the queries are often complicated and not optimized for the temporal domain. Some researchers approach this problem by extending the query language with new constructs and optimizations for temporal datasets to create new languages such as SRQL [5], and SQL-TS [6]. This is often done when the temporal dataset is represented using a relational data model. Other projects introduce new time-oriented query systems that often include interactive visual query interfaces [7], [4]. The most frequently used functional operators of the interactive querying systems are logical expressions: conjunction, disjunction, and neighboring relations.

TVQL [7] and TVQO [8] are novel direct manipulation query interfaces for specifying temporal relationships between two temporal entities using double sliders. Double sliders are used to form temporal relationships between two intervals. This direct manipulation design allows users to specify particular queries or browse to discover temporal trends. A similar layout using double sliders is employed in Patternfinder's result visualization panel.

KNAVE [4] is a system that incorporates RESUME to visualize and interactively explore temporal abstractions or raw medical data. Users can dynamically query, examine, and visualize temporal information at multiple levels of abstraction.

2.3 Visualization

Visualizations of temporal data depend on the nature of the data and what aspect is to be visualized. Some visualizations focus on periodicity [9] [12]. Other systems adopt the seminal work of Tufte [10], in which a temporal interval is displayed using a timeline: a horizontal bar on a bi-dimensional space. One axis represents the time dimension and another axis represents a different attribute. Each temporal interval has temporal relationships that are a subset of 13 cases proposed by [11] (i.e., before, after, during, contains, overlaps, over-lapped-by, meets, met-by, starts, started by, finishes, finished-by) .

2.3.1 Periodic Data

Visualizing time-series data using spirals is well suited for large data sets and aids identification of patterns and periodic structure within the data. In this approach, each ring of the spiral represents a periodic section of the data [9].

Datajewel, a system that facilitates temporal data exploration, combines two familiar visualization components into a new visualization technique, CalendarView. CalendarView represents event data on a daily basis based on the visual metaphor of a calendar and histograms are used to illustrate the frequency of events per date [12].

2.3.2 Tufte's Visualization Method

There are several approaches for visualizing time points, intervals, relations, and logical expressions [13]. Systems that adopt Tufte's [10] approach have similar conventions [7, 4, 14, 15, 8, 16, 17]. Time points and time intervals are represented using shapes such as boxes or circles. The label, height, and color capture the meta-data associated with these temporal entities. In cases where domain knowledge is also provided, qualitative information can be encoded from the quantitative data [17] and visualized. Temporal relations were specified using double sliders, arcs connecting the temporal nodes, or just by positioning boxes and circles in a relative coordinate system. [14] introduced history and relation oriented views in addition to encoding multiple variables into the timestamp.

3 PATTERNFINDER MOTIVATION AND DESIGN PARAMETERS

3.1 Sparse Time-Oriented Dataset

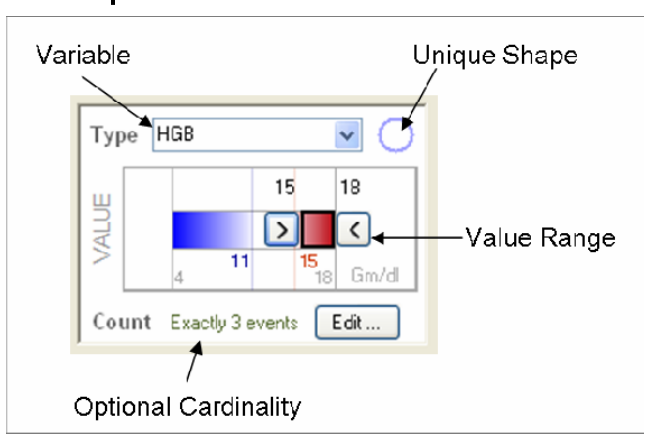


Figure 2 Event box in Visual Query Interface: constraints can be variable type or value range and cardinality.

Our dataset was extracted from a relational database with time stamps for each tuple. It is comprised of 24,819 blood test results from 552 patients over 65 years old. These tests were conducted by 53 doctors from September 6th, 2005 to October 11th, 2005 to facilitate detecting abnormal patterns among blood tests, urine tests, and cerebral spinal fluid tests. These lab test results were numerical, categorical and ordinal.

Reference ranges of normal patient conditions were provided for the numeric tests. For example, a normal person's reference level for white blood cell (WBC) count is 3.00-8.50K/ul. Patternfinder uses an encoding scheme to indicate test results above and below the reference level (Figure 2).

3.2 Three Major Components

Patternfinder includes components for modeling temporal datasets, forming queries, and visualizing temporal patterns. The data modeling component adapts an input temporal dataset to fit application's data model. The dataset should be a series of timestamps with attributes. Other data models require datasets that are composed of intervals with attributes.

The visual query interface aims to present an interaction paradigm that supports the required tasks. In the medical domain, monitoring patients' conditions and reporting daily abnormalities is a challenge. Often, queries are hard coded into an alert system because it is difficult for users to express relational database queries with temporal aspects, and because only a subset of the data is stored in the memory. However, a clinical system should meet the changing data needs of the clinician and the clinician-manager [18]. Systems should be able to perform common management-related data tasks along with common single-patient clinical data tasks [16].

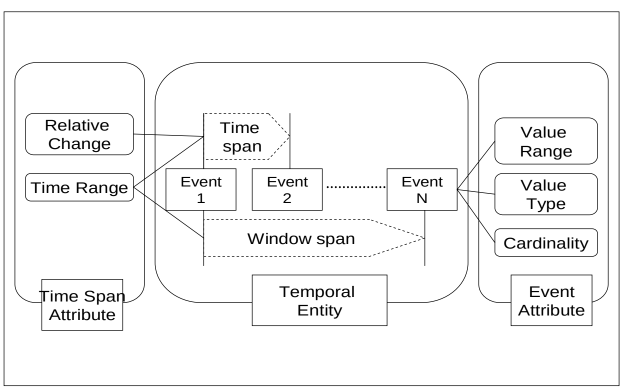


Figure 3 Temporal Data Model: (Top) Original spreadsheet data set can be visualized using table tab that gives coordinated window view with the result visualization panel. (Bottom) Temporal entities are Time Span, Window Span and Event. Temporal attribute are constraints that user can assign to the temporal entities. A sequence of events and time spans is called a *Pattern*.

Finally, the visualization component aims to intuitively reveal temporal attributes of the dataset and display both overview and detail information about patterns returned from queries formed using the visual query interface.

3.3 Task Domain

Patternfinder distinguishes between three categories of tasks which are common in the medical domain: event search, relative change search and sequence search. These are described in Figure 5. Previous versions of Patternfinder used a three-level hierarchy to organize events into groups. This hierarchy was included in the query formulation interface. A usability study using the dataset described in section 3.1 determined that the hierarchy was unnecessary in most instances, and suggested some improvements to the visual layout of the query formulation interface.

3.4 Implementation

Patternfinder was originally implemented as a desktop application using Visual C#.NET and the Piccolo.NET toolkit [19]. It is being converted into an ActiveX control so that it can be integrated into the system at Washington National Hospital, where it can be used by medical staff to monitor patient histories.

Queries are evaluated using a sequential scan of data fields that correspond to the data model (Figure 3) residing in memory. In order to support dynamic queries all event information is cached in memory. Therefore, although the query set that Patternfinder supports can be issued to any commercial relational database and be visualized in the visualization panel, this is left to future work.

4 DATA MODEL

4.1 Temporal Entity

A temporal entity can be a time stamp or a time interval, which correspond to *Events* and *Time spans respectively.* Time spans can be expressed by two events and a *Pattern* is a sequence of events and time spans (Figure 3).

*Event*s are characterized by two attributes: its variable type and its value range. For example, a HGB blood test result with a range of 10~14 is an event where HGB blood test is the variable type and 10~14 is the value range. There may be several tuples in the dataset that are a subset of this type of event.

Event Search	→ Find all patients who have not had a pep smear for the last 3 years → Find all patients who received an emergency department diagnosis of thrombophlebitis since the beginning of the year → Find all patients who have a HGB (Hemoglobin) measurement that is higher than the 14mmHg upper reference level.
Relative Change Search	*An increase in BNP (Brain natriuretic peptide) is a sign of worsening congestive heart failure. (Roughly a 50% increase would be considered clinically significant - from 300 to 500 would be of mild concern.)* → Find all patients with more than 50% increase of BNP. *An increase in creatinine is a harbinger of kidney failure. An increase in 0.5 would be considered something to watch closely. An increase in 1.0 would be of great concern.* → Find all patients who have an increase of 0.5 in the creatinine indicator.
Sequence Search	→ Find all patients receiving heparin whose platelet counts have dropped by 15%. → Find all patients who returned to the ED within 72 hours of a prior visit and were admitted.

Figure 5 Example temporal queries in the medical domain.

Time span is an abstract data entity that exists only the result set. Two events can define a time span of interest. Events and time span have a dual relationship because each entity can be expressed in terms of the other entity. In other words, time span can be used to express the relationship between two events and events can be used to define a time span.

Window span is a constraint that can be set on a pattern (N events) instead of between two events. For instance, if users want to set a time constraint on sequences of multiple events, window span can be used.

4.2 Timebox

There are two types of time range attribute for time span and window span. *Relative time range* indicates minimum and maximum time difference between two or more events. Using the relative time range, users can manipulate the width of a timebox around events in the resulting pattern [20]. The second type of time range is an *absolute time range* that users can use to specify the absolute position of the timebox.

Since the temporal dataset is sparse, setting the absolute position or size of the timebox using the two constraints above may lead to an empty result set. Patternfinder uses a looser constraint called the *cardinality* to set the size of the timebox based on the minimum or maximum number of events that should occur in it.

4.3 Pattern Types

Patterns can be sequence of events (E) or sequence of time spans (T). Patternfinder allows the user to query only a subset of these sequences and time spans that encompasses the task list (Figure 5).

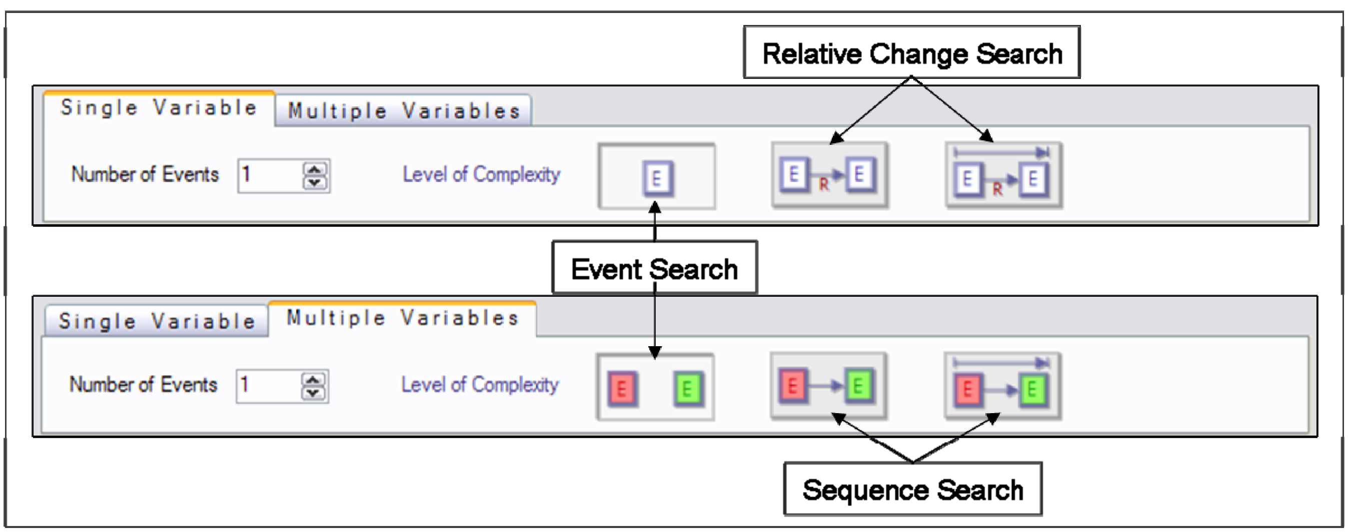


Figure 4 Visual query interface with layered approach: The *relative change search* tasks are issued mostly with single variable value and *sequence search tasks* involves multiple variable value. Under each tab, there are three levels of complexities to choose from.

4.3.1 Event Search

The simplest query users can issue is a search for an event with a specified value range. Users can also specify the cardinality of the event which is indicated by a value at the bottom of the event box (Figure 2). The following is an SQL query that expresses the following question: "*Find all patients with exactly three HGB measurements above 15 Gm/dl*". This is illustrated in Figure 2. The three constraints that are set in this simple event search are the variable, the value range, and the cardinality.

```
SELECT P.*
FROM        Person P, Event E
WHERE       E.type = "HGB"
            E.value >15
As A

SELECT P
FROM        A
WHERE       P.count =3
```

4.3.2 Single Variable Searches

Doctors are interested in monitoring their patients' conditions over time. A sample query is: "*Find all patients with more than 8mL increase in RBC (Red Blood Cell) measurement in 2 to 8 hours*". Two constraints can set in single variable searches: the relative change and the time range between these two events (Figure 6 top screenshot).

```
SELECT P.*
FROM        Person P, Event E1, Event E2
WHERE       E1.type = "RBC"
            E2.type = "RBC"
            E2.value > E1.value + 8
            (E2.time – E1.time) > 2
            (E2.time – E1.time) < 8
```

4.3.3 Multiple Variable Searches

Queries in this category are often used to examine cause and effect relationships. Users specify the sequence of events and the time spans between these events. A sample query is "*Find al patients who had an HGB test followed by an SAT blood test*" (Figure 6 middle screenshot).

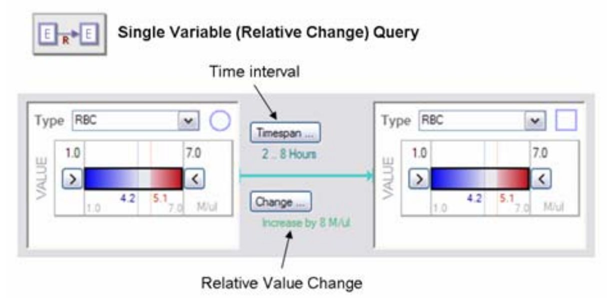
```
SELECT P.*
FROM        Person P, Event E1, Event E2
WHERE       E1.type = "HGB"
            E2.type = "SAT"
            E1.time < E2.time;
```

4.3.4 Window Span Constraint

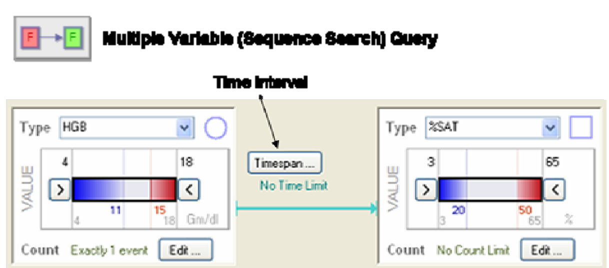
Window span constraints are often applied to more than three events. For example, clinicians may search for patients who had HGB, RBC and WBC tests where the time constraint between HGB and RBC is four to eight days, the time constraint between RBC and WBC is zero to eight days and the window constraint is zero to 12 days between first and last event (Figure 6 bottom screenshot).

5 VISUAL QUERY INTERFACE

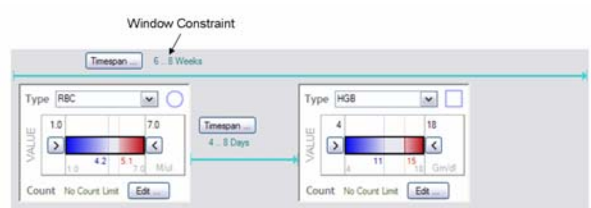
Patternfinder introduces a visual query interface that supports the user tasks described in Figure 5. This application is intended for non-technical users in the medical domain who prefer a simple intuitive interface.



Query: Find all patients with more than 8mL increase in RBC (Red Blood Cell) measurement in 2 to 8 hours.



Query: Find all who had an HGB test followed by an SAT blood test.



Query: Find all patients who had RBC test followed by HGB test within four to eight days, followed by a WBC test within zero to eight days, where the window constraint between the first and the last event is zero to twelve days.

Figure 6 Visual Query Interface. (Top) In Single Variable View, users may specify a Relative Change query by placing appropriate constraints on the spanning arrow. (Middle) In Multiple Variable View, users may search for multiple events separated by a given time interval (Bottom) A window span constraint around a sequence search.

One of its primary goals is to hide complicated aspects of query-making from users who want to conduct simple queries. To this end, Patternfinder breaks up the tasks into levels. As the user advances to higher levels, more layers of visual components are added to the interface. At higher levels the user can make more complicated queries. This kind of multilayered design results in a "level-structured interface" [21].

To break up the task space conceptually, Patternfinder takes advantage of the fact that Relative Change Search tasks involve only one variable while Sequence Search tasks may search over multiple variables. This distinction is represented using two different tabs shown in Figure 1.

Within each tab, the user chooses a level to reveal or hide more layers depending on the complexity of the desired query. The first level is a search for an event. Events are encapsulated using a simple *Value Constraint Box* (Figure 2) which allows users to choose an event and set its value range. In the multiple variable view, the user can perform this search using more than one value constraint box.

The next level is a search involving two value constraint boxes separated by a *Spanning Constraint Arrow* (Figure 6 – top, middle). In the single variable view (Figure 6 – top), this arrangement represents a simple Relative Change with the second value constraint box treated as a value increase or decrease relative to the first box. In the multiple variable view (Figure 6 – middle), this arrangement represents a Sequence Search and the two boxes can be set to different variables. In both views the spanning arrow can be used to specify a time interval between the two value constraints. Time intervals may be specified at four different granularities: minutes, hours, days, and weeks.

These first two levels will be sufficient for most users. An additional level is added to both views to allow the users to bound all their results within a time window. This window is represented using a spanning constraint arrow similar to the one in the second level, but placed around all the boxes (Figure 6 – bottom). This interface is the most complicated, but is not likely to be used often. Hiding it at the third level ensures that users perform most of their tasks with a simpler interface.

6 RESULT VISUALIZATION PANEL

The result visualization panel shows patients who have events that satisfy all of the query constraints and their corresponding events. Figure 7 shows the results of the query in Figure 6 (middle). We use Tufte's approach [10] of visualizing query results, in which the x-axis represents the timeline and the y-axis is an alphabetical list of patients who meet the query criteria.

For each person that is displayed, two types of events, 'background' and 'matching', are shown. Matching events are those that meet a query constraint and are represented

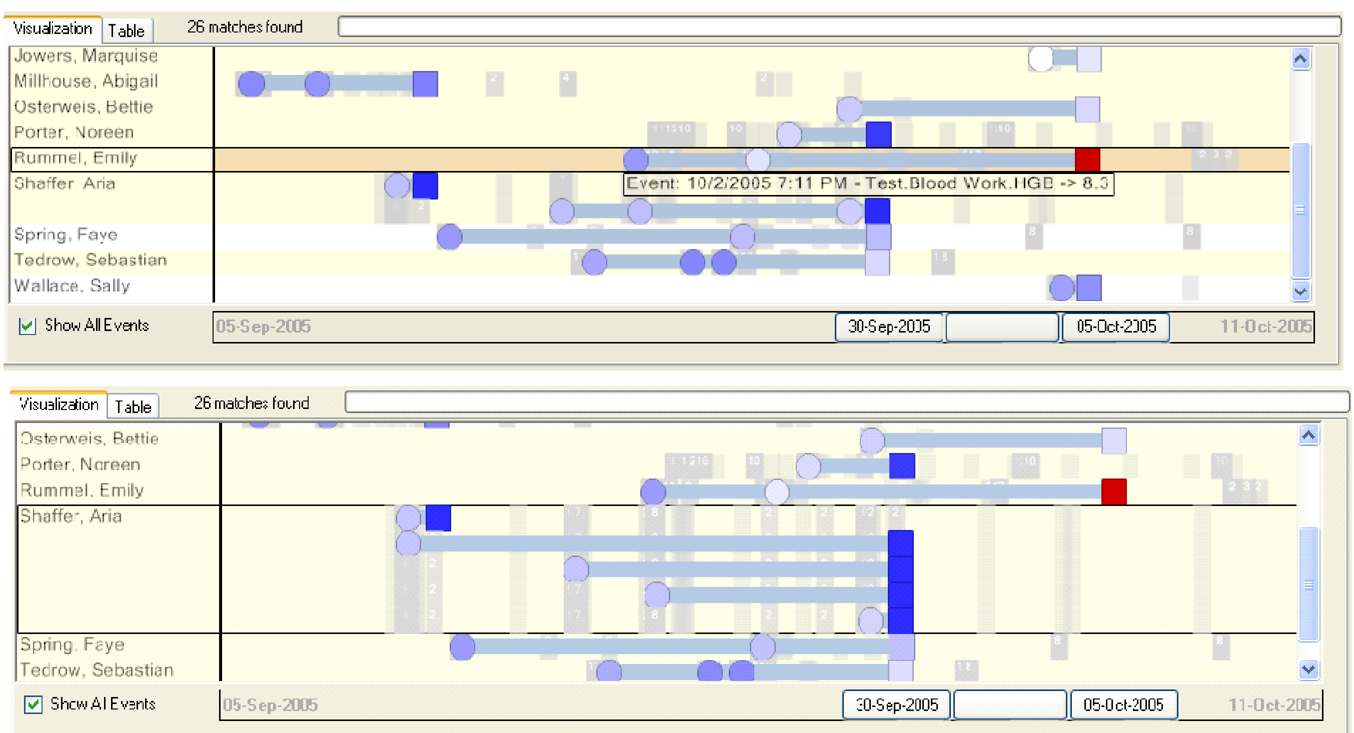


Figure 7 Query Result Visualization Panel: This result was generated using the Figure 6 (Middle) query. It is looking for patients who had SAT% test after HGB (Hemoglobin) count measurement. Aria Shaffer had a total of four HGB (circle) events with two SAT (square) event. The top panel shows a compressed view of her results. Upon clicking that row, it will expand to give details of each pair. Blue and red shapes indicate results below and above the reference level respectively.

by the unique shape (i.e., triangle, square, and ellipse) of the corresponding event box (Figure 2). For numerical queries in which the events have reference values, the color of the shape encodes the value of the event relative to the reference values. If the value is above the reference level (too high) the shape will be a shade of red, and if it is below the reference level (too low) the shape will be a shade of blue. 'Normal' values that fall within the reference levels will be white.

Background events are those that do not meet the query criteria. They are all represented by grey squares. Although not of initial interest to the user, background events may offer some insight into the query matches. For example, Figure 7 (top) shows that patient Emily Rummel had a very low HGB test followed by a slightly low HGB and extremely high %SAT test. The background events in between the two HGB tests may provide plausible cause or explanation for the change in value. Detailed information for both event types is available via a tooltip.

Unlike many similar applications [16, 22, 8, 4], PatternFinder utilizes two (as opposed to one) visual variables, shape and color, to encode event information. This allows users to get a sense of event values and temporal placement without having to consult a more detailed view, permitting users to gain more insight into the results with less effort. Accordingly, PatternFinder facilitates easier identification of patterns and trends and is useful for directed as well as exploratory data analysis.

Adopting an overview and detail technique, there are two possible views, abstract and expanded, of the result set for each patient. Figures 7 and 8 illustrate the difference. The expanded view, which can be reached via a mouse click on the person of interest, displays all possible combinations of events matching a query (Figure 8, left). The default abstract view simplifies the expanded visualization by grouping events in the order in which they occur, displaying the same information in a condensed format (Figure 8, right).

To avoid inefficient waste of white space, PatternFinder automatically adjusts the double sliders that control the date range in the visualization to zoom in on the current query results. Statistical information, such as the number of patients that match the query out of the total number of patients, and the total number of matches found is displayed in the upper right hand corner of the visualization panel.

The visualization panel also contains a table view (Figure 9) which is tightly coupled with the visualization view. The table view provides detailed information about each patient, including the number of events they have and the number of those events that match the current query.

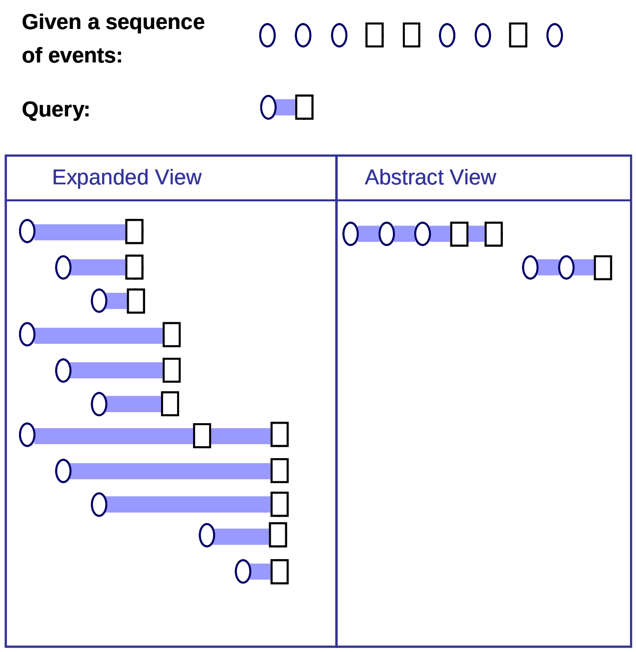
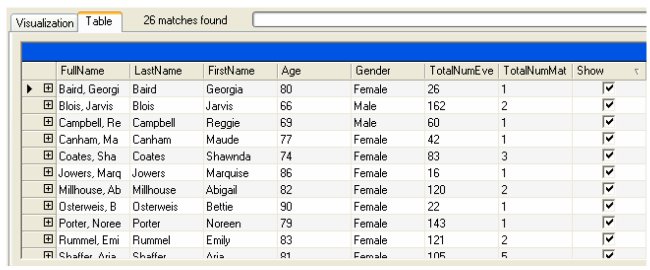


Figure 8 Given the illustrated sequence of events and query, this shows the expanded and abstract view of the results.



Visualization | Table | 26 matches found

	FullName	LastName	FirstName	Age	Gender	TotalNumEve	TotalNumMat	Show
▸ ⊞	Baird, Georgi	Baird	Georgia	80	Female	26	1	☑
⊞	Blois, Jarvis	Blois	Jarvis	66	Male	162	2	☑
⊞	Campbell, Re	Campbell	Reggie	69	Male	60	1	☑
⊞	Canham, Ma	Canham	Maude	77	Female	42	1	☑
⊞	Coates, Sha	Coates	Shawnda	74	Female	83	3	☑
⊞	Jowers, Marq	Jowers	Marquise	86	Female	16	1	☑
⊞	Millhouse, Ab	Millhouse	Abigail	82	Female	120	2	☑
⊞	Osterweis, B	Osterweis	Bettie	90	Female	22	1	☑
⊞	Porter, Noree	Porter	Noreen	79	Female	143	1	☑
⊞	Rummel, Emi	Rummel	Emily	83	Female	121	2	☑
⊞	Shaffer, Aria	Shaffer	Aria	81	Female	105	5	☑

Figure 9 Table view of the results of query in Figure 6 (middle).

7 FUTURE WORK

Patternfinder has limitations in its input data model and its ability to efficiently handle large datasets or complex queries. Its input is limited to a static flat file prepared ahead of time. Temporal datasets are often made up of real-time events updated regularly.

Real time datasets can get very large. Patternfinder stores its dataset in memory to support dynamic queries. This may not always be practical and the system may need to decide how much data to cache in order to perform optimally.

Queries involving many events and time spans can become quite complex. Patternfinder's sequential search approach is slow in some cases and some queries even cause the system to crash. Research is needed to find query optimization techniques and processing algorithms [23] that enhance its performance.

Finally, extensive long-term user studies involving real users in the medical domain may help identify areas for improving the interface and supporting the needs of its users.

8 CONCLUSION

Pattern search using Patternfinder in the medical domain is useful for searching for events, searching for changes in a variable's value and discovering causal relationships between events. Patternfinder provides a visual interface for forming queries over temporal datasets that are composed of timestamps with attributes. There are other temporal data models that can be explored, but Patternfinder's data model is the most appropriate for most medical applications. Patternfinder uses an intuitive overview to visualize its results, allowing users to quickly observe temporal patterns.

Users in the medical domain with a limited amount of time to learn the application can easily form queries for most commonly performed medical tasks on patient history data and visualize the results ordered by the patient's name on a timeline.

Patternfinder is focused on the medical domain and is not intended as a general purpose query interface that connects to commercial databases. Nevertheless, its approach can be applied to other sparse temporal domains.

9 REFERENCES

[1] Carlo Combi and Yuval Shahar, Temporal Reasoning and Temporal Data Maintenance in Medicine: Issues and Challenges, Computers in Biology and Medicine. 1997

[2] Fries JF. Time-oriented patient records and a computer databank. JAMA. 1972;222(12):1536-42

[3] Y. Shahar A Knowledge-Based Method for Temporal Abstraction of Clinical Data. Diss. Stanford University. 1994

[4] Y. Shahar and C. Cheng. Model-based visualization of temporal abstractions. In L. Khatib and R. Morris, editors, Proceedings of the Fifth International Workshop on Temporal Representation and Reasoning (TIME98), pages 11--20. IEEE Computer Society Press, 1998. http://citeseer.ist.psu.edu/shahar98modelbased.html

[5] R. Ramakrishnan et al. SRQL: sorted relational query language. Statistical and Scientific Database Management, 1998, 84-95.

[6] Reza, S., Zaniolo, C., Zarkesh, A., and Adibi, J. Optimization of Sequence Queries in Database Systems. Symposium on Principles of Database Systems, 2001.

[7] Stacie Hibino, Elke A. Rundensteiner, User Interface Evaluation of a Direct Manipulation Temporal Visual Query Language, ACM Multimedia '97 Conference.

[8] S. Fennandes Silva, U. Schiel, and T. Catarci, "Visual Query Operators for Temporal Databases", In Proceedings fourth intemational workshop on temporal representation and reasoning (TIME '97), IEEE Computer Press, Los Alamitos

[9] Weber, M., Alexa, M. and Muller, W.(2001).Visualizing Time Series on Spirals. *Proc. 2001 IEEE Symposium on Information Visualization*. San Diego, CA, Oct 21-26., 7-14.

[10] The Visual Display of Quantitative Information, Graphics Press, Cheshire, 1982.

[11] Allen, J.F. Maintaining Knowledge About Temporal Intervals. CACM, 26(11), 832-843

[12] Ankerst, M., Jones, D.H., Kao, A. and Wang, C. (2003). DataJewel: Tightly integrating visualization with temporal data mining, *Proc. of DM Workshop on Visual Data Mining.*, ANSI (1986). The Database Language SQL, Document ANSI X3.135-1992.

[13] Representation of Temporal Intervals and Relations: Information Visualization Aspects and their Evaluation.

[14] C. Combi, L. Portoni, and F. Pinciroli, "Visualizing Temporal Clinical Data on the WWW", In Proceedings Joint European Conference on Arti3cial Intelligence in Medicine and Medical Decision Making (AIMDM'99), Springer Verlag, LNAI1620, Berlin, 1999, pp. 301-311.C.

[15] R. Kosara and S. Miksch, "Metaphors of Movement: A Visualization and User Interface for Time-Oriented, Skeletal Plans", to appear in Am3cial Intelligence in Medicine, Special Issue: Information Visualization in Medicine, 2001.

[16] C. Plaisant, B. Milash, A. rose, S. Widoff, and B. Shneiderman, "LifeLines: Visualizing Personal Histories", In Proceedings of the CHI '96 Conference on Human Factors in Computing Systems, ACM Press, New York, 1996, pp.221-227

[17] Bade, R., Schlechtweg, S., and Miksch, S. Connecting time-oriented data and information to a coherent interactive visuallization. CHI '04. ACM Press, New York, NY, 105-112.

[18] Fried, Craig et al, Clinical Information Systems: Instant Ubiquitous Clinical Data for Error Reduction and Improved Clinical Outcomes, ACADEMIC EMERGENCY MEDICINE2004 http://www.imedi.org/dataman.pl?c=lib&dir=docs/Azyxxi

[19] Bederson, B. B., Grosjean, J., & Meyer, J. (2004). Toolkit Design for Interactive Structured Graphics, *IEEE Trans. on Software Engineering*, 30 (8), pp. 535-546.

[20] Jensen, C.S. and Snodgrass, R.T. (1999). Temporal Data Management. *IEEE Transactions on Knowledge and Data Engineering*, 11 (1). 36-44.

[21] Plaisant, Catherine. Information Visualization and the Challenge of Universal Usability. *In Exploring Geovisualization* (Eds, J. Dykes, A. MacEachren and M.J. Kraak), Oxford: Elsevier (2005), p. 53-82. (HCIL-2004-36).

[22] Mamykina, L., Goose, S., Hedqvist, D., and Beard, D. V. 2004. CareView: analyzing nursing narratives for temporal trends. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems* (Vienna, Austria, April 24 - 29, 2004). CHI '04.

[23] Chaudhuri, S. 1998. An overview of query optimization in relational systems. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (Seattle, Washington, United States, June 01 - 04, 1998). PODS '98.