Visual Specification of Queries for Finding Patterns in Time-Series Data

Harry Hochheiser
Department of Computer Science &
Human-Computer Interaction Lab
University of Maryland
College Park, Md 20742
+1 301 405 2725
hsh@cs.umd.edu

Ben Shneiderman
Department of Computer Science,
Human-Computer Interaction Lab,
Insitute for Advanced Computer Studies, &
Institute for Systems Research
University of Maryland
College Park, MD 20742
+1 301 405 2680
ben@cs.umd.edu

ABSTRACT

Widespread interest in discovering features and trends in time- series has generated a need for tools that support interactive exploration.This paper introduces timeboxes: a powerful graphical, direct-manipulation metaphor for the specification of queries over time-series datasets. Our TimeFinder implementation of timeboxes supports interactive formulation and modification of queries, thus speeding the process of exploring time-series data sets and guiding data mining. TimeFinder includes windows for timebox queries, individual time-series, and details-on-demand. Other features include drag-and-drop support for query-by-example and graphical envelopes for displaying the extent of the entire data set and result set from a given query. Extensions involving increased expressive power and general temporal data sets are discussed.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces: -*Graphical User Interfaces, Interaction Styles* H.2.8 [**Database Management**] Database Applications *Data Mining.*

General Terms

Human Factors, Design

Keywords

Information Visualization, Time-Series Data, Data Mining, Graphical User Interfaces

1. INTRODUCTION

Interest in time-series data has prompted a substantial body of work in the development of strategies for storing and indexing temporal data. Furthermore, algorithmic and statistical methods for identifying patterns [1,2,3,7,10] have provided substantial functionality in a wide variety of situations.

Algorithmic research only addresses one aspect of the data mining problem. The question of *query formulation* – which questions are worth asking? – is often left unanswered. For example, problems involving identification of series that are similar to a known series involve specification of several query parameters. In addition to the input query, users must provide parameters defining the range of allowable similarity. Identifying of parameters such as these using trial-and-error processing is often challenging and computationally expensive. A central problem for user is that the effects of small changes on parameters such as similarity tolerances may be hard to gauge without running multiple trials.

In these cases, users need tools to support interactive exploration of the contents of time-series data sets. By providing analysts with the power to quickly construct queries, modify parameters, and examine result sets, these tools encourage the development of understanding of the data set as a whole. This understanding is useful for guiding the construction of queries, thus speeding the process of knowledge discovery.

Dynamic queries [4] and related information visualization techniques [9] have proven useful in supporting users interested in understanding multi-dimensional abstract datasets. The combination of graphic displays with easily manipulated user-interface widgets for query formulation allows users to explore data sets in search of items of interest. Although there has been little work to date on interactive systems for querying time-series data, lessons from information visualization research can guide developers of systems for the exploration of time-series data sets.

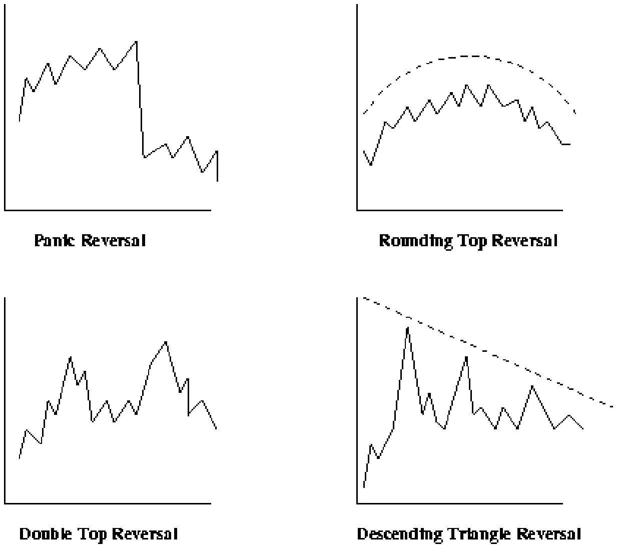


Figure 1: Patterns of interest in stock trend analysis [1]

The existence of familiar graphic displays of time-series presents an obvious starting point for the application of information visualization techniques. Two-dimensional graphs with time on the x-axis, and a continuous variable on the y-axis are ubiquitous: stock charts, weather data, and physiologic data (EEG, EKG) etc., are just a few examples. In domains such as stock price analysis, familiar patterns have been named and identified as shorthand approaches to identifying trends of interest (Figure 1) [18].

This paper introduces *timeboxes:* an interactive mechanism for specifying queries on temporal data sets. timeboxes are rectangular regions that are placed and directly manipulated on a timeline, with the boundaries of the region providing the relevant query parameters. After introducing related research, we provide an overview of the timebox model, along with a discussion of TimeFinder, a data exploration system based on the timebox query model. The functionality of timeboxes is described, particularly with respect to prior work on time series data. Finally, we discuss planned extensions that will expand the expressive power of timebox queries.

2. RELATED WORK

2.1 Data Mining

The data mining community has developed a wide variety of innovative techniques for algorithmically extracting interesting patterns from time series. Useful approaches including dynamic time warping [7] and Discrete Fourier Transforms (DFT) in combination with spatial queries [10]. To date, the focus in data mining work has been on the development of search algorithms, with relatively little attention to query specification or interactive systems. One exception is Agrawal et al.'s Shape Definition Language, which specifies queries in terms of natural language descriptions of profiles (e.g., "(zero appears up up down)") [1]. Although a user interface is not described, an interactive system might support creation of queries through combinations of Shape Definition Language primitives. Support for progressive refining of queries was addressed by Keogh and Pazanni, who suggested the use of relevance feedback for results of queries over time series data [17].

Our work with timeboxes is aimed at developing tools to address issues of user interaction with these data mining tools. TimeFinder handles many of the queries that may be of interest to the data mining community, and we believe that it offers a model to express even more complex queries. Furthermore, in supporting interactive query specification and modification, timeboxes may be particularly useful for queries involving approximate matching [21], time warping [7,27], translation [3], and outlier detection [16].

2.2 Information Visualization

Traditional time-series graphs are among the most familiar data displays. Visualizations of time-series data attempt to improve the utility of these common graphs, through the use of techniques such as increased data density or polar-coordinate displays that emphasize the serial periodic nature of the data set [8], or by distorting the time axis to realize denser information displays [20]. A recent survey of linear temporal visualizations is found in [23]. Generally, these tools focus on visualization and navigation, with relatively little emphasis on querying data sets.

A few tools have been developed for querying time-series data. MIMSY [12] provided an early example of searches for temporal patterns in stock market data, using text entry fields, pull-down menus, and other traditional widgets to specify temporal constraints. QuerySketch is an innovative query-by-example tool that uses an easily-drawn sketch of a time-series profile to retrieve similar profiles, with similarity defined by Euclidean distance [26]. Although the simplicity of the sketch interface is appealing, the use of Euclidean distance as a metric can result in non-intuitive results [17].

Spotfire's Array Explorer 3 [24] supports graphically editable queries of temporal patterns, but the result set is generated by complex metrics in a multidimensional space. This potent approach produces useful results, but users may wish to constrain result sets more precisely. For example, retrieving stocks that trade in a narrow range (up or down by no more than $2) for three months and then rise at least $10 in each of the next three months.

3. TIMEBOXES: INTERACTIVE TEMPORAL QUERIES

Timeboxes are rectangular query regions drawn directly on a two-dimensional display of temporal data. The extent of the Timebox on the time (x) axis specifies the time period of interest, while the extent on the value (y) axis specifies a constraint on the range of values of interest in the given time period. We assume that the temporal data is divided into discrete time points of granularity determined by each data set. More specifically, a timebox that goes between ($x_{\min}$, $y_{\min}$) and ($x_{\max}$, $y_{\max}$) indicates that for the time range $x_{\min} \leq t \leq x_{\max}$, the dynamic variable y must have a value in the range $y_{\min} \leq y \leq y_{\max}$.

Creation of timeboxes is straightforward: the user simply clicks on the desired starting point of the time box and drags the pointer to the desired location of the opposite corner. As this is identical to the mechanism used for creating rectangles in widely-used drawing programs, this operation should be familiar to most users. Once the timebox is created, it may be dragged to a new location or resized via appropriate resize handles on the corners (Figure 2), using similarly familiar interactions.

In all cases, query processing occurs on mouse-up. When the user releases the mouse, the current position of the timebox is stored, the query is updated, and the new result set is displayed.

Timeboxes are constrained to occupy an integral number of time points, thus enforcing the discrete nature of the data. Multiple timeboxes can be combined to specify conjunctive queries. Data sets must match all of the constraints implied by the active timeboxes in order to be included in the result set.

The combination of simple, familiar idioms for creation and modification of time boxes with automatic query processing on mouse-up provides an interactive environment suitable for data exploration. As queries are constructed and evaluated quickly (100ms) and easily, users are not constrained by the need to adequately and completely describe queries in advance [4,22]. Instead, they can easily and quickly try a wide range of queries, modifying these queries to quickly see the effects of changes in query parameters. This ability to easily explore the data is helpful in identifying specific patterns of interest, as well as in gaining understanding of the data set as a whole.

An example query involving monthly stock prices over the course of 13 months shown in Figure 2. The timebox on the left restricts the query to stocks that had low prices (between $16 and $125) during September and October, while the other timebox restricts results to stocks that had relatively high prices (between $120 and $254) in December and January. Taken together, these elements form a query for stocks that started at a (relatively) low price range and moved to a higher price range.

Additional timeboxes can be added to specify searches for more complex patterns: Figure 3 provides an example timebox query for panic reversal patterns similar to the example given in Figure 1. In addition to being succinct and easy to create, the timebox version of this query provides a visual picture of the pattern that is not apparent in other notations. For example, the query in Figure 3 is more easily interpreted than the mathematical expression of the same constraints, which is cognitively more difficult for users to comprehend (Table 1).

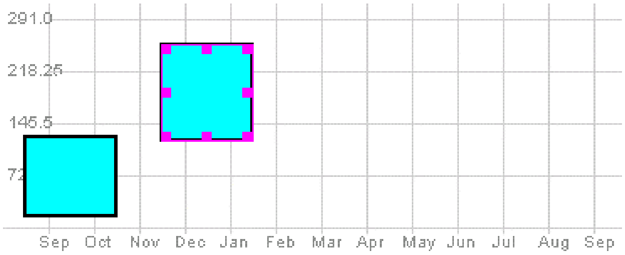


Figure 2: A conjunctive timebox query, with resize handles shown on the right-hand timebox

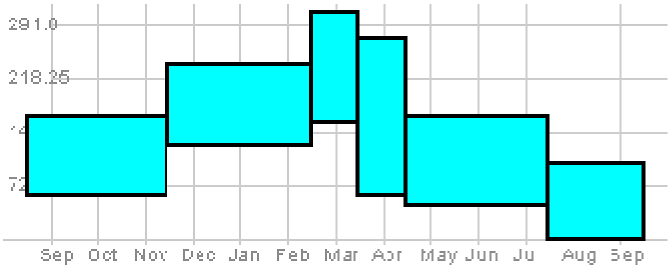


Figure 3: timebox query for Panic Reversal pattern

$57 \le y \le 160$ when	$sep \le t \le nov$
$124 \le y \le 230$ when	$dec \le t \le feb$
$154 \le y \le 291$ when	$t = mar$
$58 \le y \le 266$ when	$t = apr$
$46 \le y \le 162$ when	$may \le t \le jul$
$0 \le x \le 101$ when	$aug \le t \le sep$

Table 1: Panic reversal query constraints

Although their development was motivated by interest in querying time series data sets, timeboxes are more general. For example, any ordinal dimension might be used on the x axis, thus providing opportunities to query sequential data such as DNA sequences, or any general function $f(x)$. Enhancements include interpreting the y axis as describing discrete categories. This modification supports the use of timeboxes for searching over categorical data sets such as those found in medical records [19,20].

4. TIMEFINDER

4.1 Overview

Our TimeFinder software uses timeboxes to pose queries over a set of entities with one or more time-varying attributes. Entities have one or more static attributes, and one or more time-varying attributes, with the number of time points and the definition of those points being the same for every entity in a given data set. If there are multiple time-varying attributes, any one of them can be selected for querying, through a drop-down menu which specifies the dynamic attribute being queried. All active queries refer to the same attribute. Our initial data set involves monthly low and high prices for a set of stocks, over the course of 13 months. We have also used TimeFinder with data sets tracking one month of daily prices or one year of weekly prices for over 1400 stocks.

When a data set is loaded, entities in the data set are displayed in a window in the upper left-hand corner of the application. Each entity is labeled with its name, and the values of the active dynamic attribute are plotted in a line graph. Complete details about the entity (details-on-demand) can be retrieved by simply clicking on the graph for the desired entity: this will cause the relevant information to be displayed in the upper right-hand window (Figure 4).

4.2 Query Creation

The bottom-left corner of the TimeFinder window is the query input space. This space initially contains an empty grid. To specify a query, users simply draw a timebox in the desired location. Query processing begins as soon as users release the mouse, signifying the completion of the box. Thus, users do not need to press a button to explicitly start a search. When query processing completes, the display in the top half of the application window is updated to show those entities that match the query constraints. For all of these entities, the time points that correspond to the queries are highlighted, in order to simplify interpretation of the display.

4.3 Query Modification

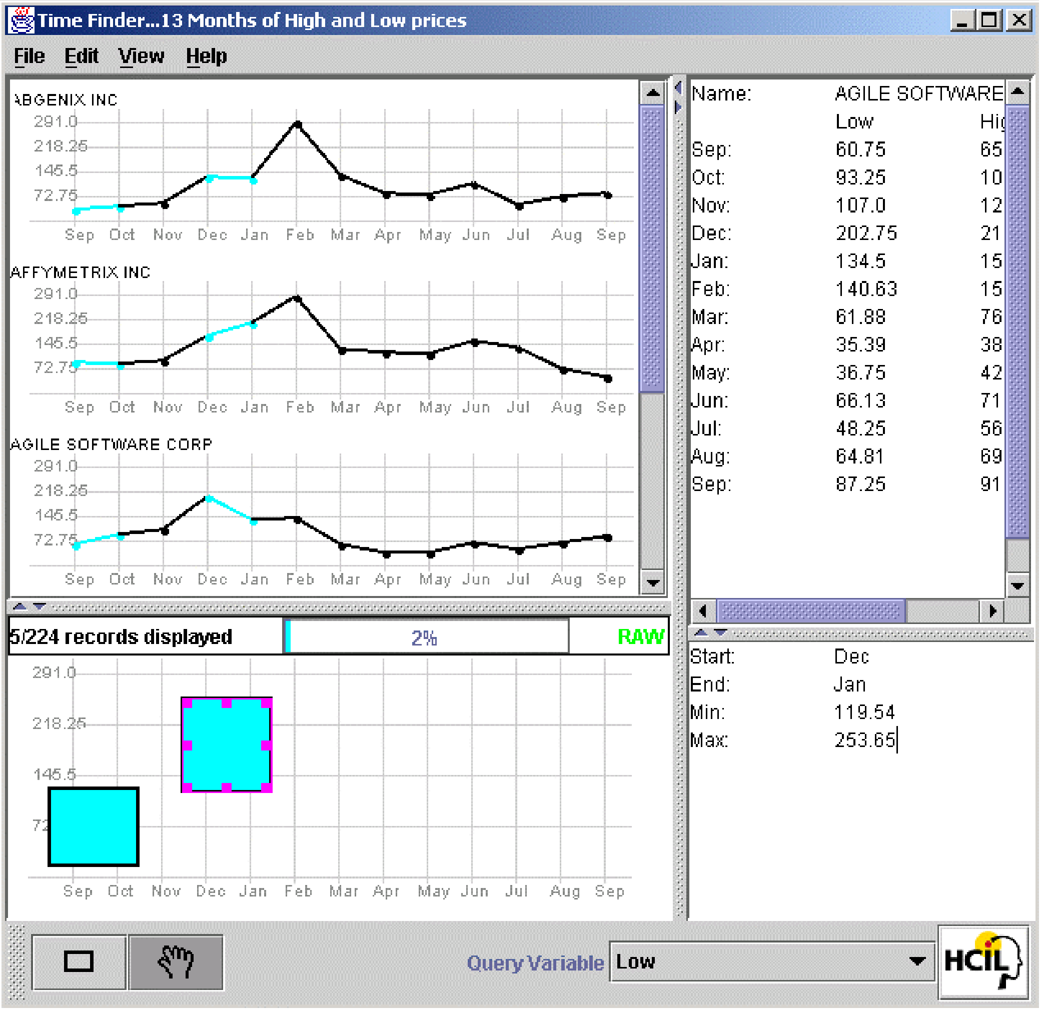
Once the initial query is created, the timeboxes can be moved and resized. The hand and box icons on the lower toolbar are used to switch between creating timeboxes and moving/resizing them. As is the case with initial timebox creation, query processing begins immediately upon completion of the movement/resizing of the timebox.

When multiple timeboxes are present, they can be modified individually or simultaneously in groups of two or more. This functionality is particularly useful for searches for complex patterns (Figure 3). In these cases, users can select some or all of the timeboxes (using standard lasso and shift-click interactions) and simultaneously apply the same translation and/or scale along either or both axes to all selected timeboxes. This is useful for searching for instances of a pattern that vary slightly in scale or magnitudes, or for modifying queries based on example items (Section 4.5 below).

The lower right-hand corner of the TimeFinder window provides details describing individual timeboxes. When the user clicks on a timebox, the details of its extent are displayed in this window. Similar details are available through tool tips, which are displayed if the mouse dwells on a timebox.

4.4 Query Guidelines

For ease in creating and modifying queries, TimeFinder can (optionally) provide *query guidelines:* additional guiding lines that extend from each of the four corners of the timebox back to the 2 axes,



5/224 records displayed 2% RAW

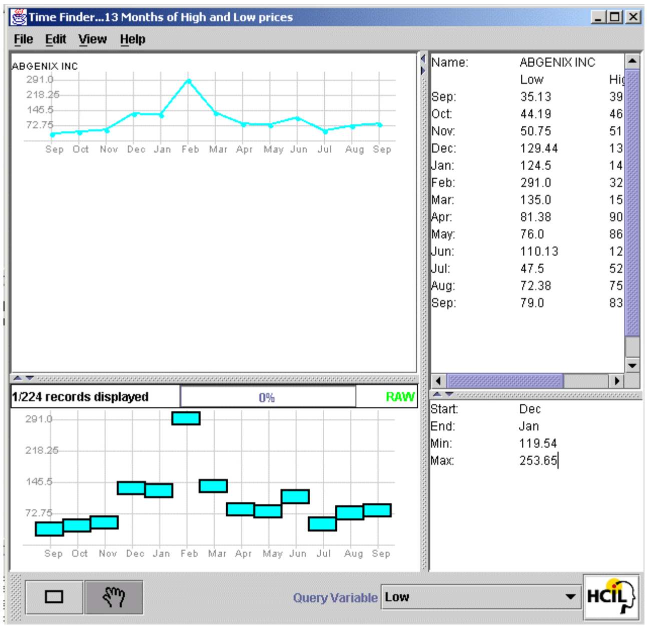
Name:	AGILE SOFTWARE	
	Low	Hi
Sep:	60.75	65
Oct:	93.25	10
Nov:	107.0	12
Dec:	202.75	21
Jan:	134.5	15
Feb:	140.63	15
Mar:	61.88	76
Apr:	35.39	38
May:	36.75	42
Jun:	66.13	71
Jul:	48.25	56
Aug:	64.81	69
Sep:	87.25	91

Start:	Dec
End:	Jan
Min:	119.54
Max:	253.65

Query Variable: Low

Figure 4: TimeFinder, displaying the query shown in Figure 2, with two timeboxes and four of the five records in the result set

indicating the extent of the box on each of the axes. Two lines go from the leftmost corners of the box, extending to the y-axis, while the other two go from the bottom corners to the x-axis. These lines move with the box as it is created or adjusted, proiding users with feedback appropriate for fine-tuning query extents.



1/224 records displayed

Name:	ABGENIX INC	
	Low	Hi
Sep:	35.13	39
Oct:	44.19	46
Nov:	50.75	51
Dec:	129.44	13
Jan:	124.5	14
Feb:	291.0	32
Mar:	135.0	15
Apr:	81.38	90
May:	76.0	86
Jun:	110.13	12
Jul:	47.5	52
Aug:	72.38	75
Sep:	79.0	83

Start: Dec
End: Jan
Min: 119.54
Max: 253.65

Query Variable: Low

Figure 5: Drag-and-Drop query-by-example

4.5 Drag and Drop

Users might be interested in identifying entities that have profiles similar to a given template or example from the data set. TimeFinder provides a simple drag-and-drop mechanism for these “query-by-example” queries: the user can simply click on an entry in the data display window, drag it into the query window, and release the mouse to drop, thus instantiating a query.

The query resulting from a drag and drop has a separate timebox for each time point in the data set. Each timebox has a width of one interval, with the query values centered around the actual value of the attribute for that entity at the given time point. The height of each timebox is set to be 10% of the total range of the attribute being queried, so each timebox has a range of $y \pm 5\%$ of the total range in the attribute value, where y is the value of the query attribute at the given time point (Figure 5).

4.6 Envelopes for Overviews

Information visualization environments generally provide overview displays to help users make sense of large data sets [9,22]. Displays that show the entire data set can help users understand global trends in the data, identify outliers, and plan query strategies. When this is not possible, an alternative is to present a simplified representation, containing some of the relevant summary information.

TimeFinder provides a limited overview display in the upper left-hand window, displaying each of the entities in the data set in a linear list. As this display shows a small number of items at any given time, it is not an effective overview. Another possible overview would display each of the entities in a thumbnail graph. These thumbnails would be displayed in a grid, instead of the linear arrangement shown in Figures 4 and 5. This approach suffers from two shortcomings. For any reasonably-sized data set (more than a few dozen items), the limited screen space available would restrict each thumbnail to a tiny area of the screen, rendering it virtually unreadable. Furthermore, displaying each entity in a separate graph may not help users in identifying global trends, such as the extreme values of the time-varying attribute at any given point in time.

The *data envelope* is an alternative overview representation that overcomes these difficulties. Optionally shown in the background of the query window, the data envelope is a contour that follows the extreme values of the query attribute at each point in time, thus displaying the range of values that may be queried (Figure 6). When the user executes a query, the data envelope is extended by a *query envelope* – an overlay that outlines extreme values of the entities in the result set (Figure 7). This display provides users with a graphic summary of the relationship between the result set and the data set as a whole.

The data and query envelopes provide users with feedback regarding appropriate strategies for creating, interpreting, and extending queries. When starting without any timeboxes present, the data envelope highlights areas that would be fruitful for query creation, while leaving empty areas unmarked. For example, the data envelope in Figure 6 does not extend to the upper left-hand corner, so queries in that region would not return useful results. When a timebox is created, the updated query envelope shows the differences between the current result set and the data set as a whole, thus clarifying the range of values excluded by the timebox. The query envelope also guides the creation of additional timeboxes, as queries outside this envelope will not match any records.

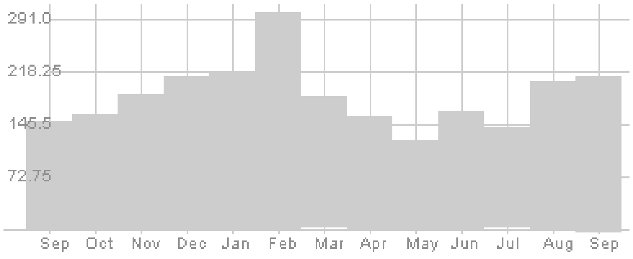


Figure 6: Query Window with Dat a Envelope

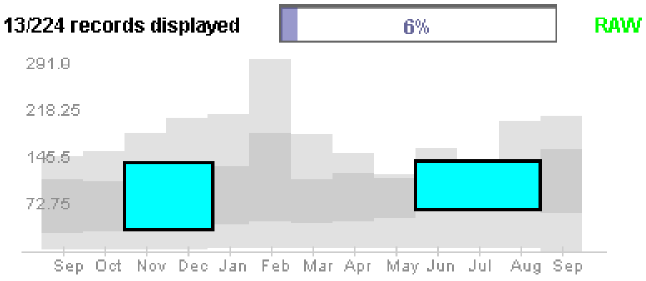


Figure 7: Query Display with Data and Query Envelopes

Additional overview information is given in the status line above the query area. This display contains a text label indicating the number of records that match the current result set, a percentage bar indicating the portion of the data set included in the query, and a text label indicating the treatment of the data (raw or normalized).

5. SOFTWARE

TimeFinder was implemented in Java 2, using the Swing toolkit for user-interface widgets. Although Java's flexibility comes with a performance penalty, our immediate concern is in the definition of appropriate widgets and interaction styles for timebox queries.

Drawing and scenegraph control in the data and query displays, along with functionality for moving and rescaling timeboxes, is provided by Jazz – a zooming toolkit written in Java and developed in the Human-Computer Interaction Laboratory at the University of Maryland [6]. Timeboxes, graphs of each item, and query and data envelopes are implemented as Jazz widgets. As these widgets are implemented as Java classes, they are easily extended for specialized use.

In addition to supporting the basic scenegraph creation and modification, Jazz also supports zooming for viewing of items at different scales. In future versions of TimeFinder, this functionality will be used to provide facilities for zooming to view a portion of a time-series graph in more detail. Alternatively, for longer time-series, Jazz's zooming could provide distorted, "fish-eye" views that would provide detail and context at differing scales [11].

As currently implemented, TimeFinder uses a simple approach to data storage and querying. Temporal data is stored in a flat file and read into RAM. Queries are processed by simply iterating over the entities, with optimizations possible for several query cases. For example, when a timebox is added to the query input space, the natural conjunctive shortcut of only checking those entities that satisfied the constraints of any previously existing query elements will significantly speed multiple-box queries. Similar optimizations are possible when a query element is deleted. Furthermore, rendering is interleaved with querying – items are rendered as they are evaluated, instead of waiting until query processing is complete.

Although the combination of flat-file storage and linear query processing works well for small data sets, larger data sets will require improved strategies. Storage requirements will obviously increase linearly with both number of items in the dataset and the number of time points, while query processing time in the current model is linear in the size of the dataset and the complexity of the query. String search algorithms suggest immediately possible speedup to sublinear times.

Planned extensions to the query facilities require alternative data structures and/or algorithms. For example, we will support queries that specify patterns that can happen at any points in time, as opposed to at some fixed time point. We will implement efficient algorithms derived from string-searching approaches. As we move into more complex queries and larger data sets, alternative data structures such as the use of spatial embedding are planned [10]. The specific choices made in these areas are likely to be determined by the problems and domains that we hope to address.

6. DISCUSSION

For tasks involving retrieval of information from large data sets, query formulation is a crucial step in the process. Insufficiently precise queries often lead to empty result sets ("zero-hit queries") [13] or large result sets with potentially irrelevant results. This problem is particularly troublesome in data mining applications where query execution is computationally expensive. Strategies that increase query effectiveness will help reduce this overhead cost, thus streamlining the knowledge discovery process.

Research in information visualization has led to the development of the "overview first" strategy [22]. Starting from a view that describes the data set as a whole, the user manipulates interface controls to specify constraints on the items of interest, thus reducing the size of the active data set. Real-time, interactive query processing and display update provide the user with feedback, as the items that are filtered out are removed from the display. Filter controls should be easily manipulated and modifiable, to encourage examination of alternatives. As actions throughout this process are low-cost and provide continuous feedback, users are never far from some data of interest (no "zero-hit" queries). If a given display contains too many items of interest, additional filters are added to reduce the result set to a manageable size (no "million-hit" queries).

TimeFinder uses timeboxes to implement this model for time-series data. The data and query envelopes, together with the linear list of graphed elements, provide the necessary overview. In particular, the envelopes plainly indicate where the user should *not* bother to place a new timebox. Each timebox is a new filter that restricts the data set resulting from the query formed by the pre-existing timeboxes. Query processing on mouse release follows a model familiar to users of modern GUIs, whereby a mouse release is treated as completion of user input.

Extending this model to support a wider-range of realistic data mining problems involves challenges in increased query expressiveness and query processing.

6.1 Query Expressiveness

For many interesting time-series questions, the exact time of the occurrence of an event of interest is less important than its relationship in time to other events, or perhaps even than its mere occurrence. As currently implemented, TimeFinder supports queries that involve fixed, specified points in time or attribute values.

Users might be interested in events of fixed duration occurring at any point in time. Events of arbitrary duration, or events separated by arbitrary time gaps may also be of interest. Similar queries for values of the time-dependent attribute are also desirable.

The timebox model is easily extendable temporal data sets, such as those found in video applications [14], medical data [19], market-basket data [2], and other applications. In these cases, events have arbitrary duration, and may occur simultaneously. Users may want to formulate queries that co-occur, follow or precede each other, or have other time relationships similar to those specified by Allen [5].

These extensions to the timebox model involve additional mechanisms for specification of queries and presentation of the results. Specifically, additional widgets for expressing an allowable range of time or attribute values will be needed, along with visualizations that clearly indicate the relationship between queries and results.

Finally, data sets often have one or more static attributes that may be used in queries: dynamic query widgets [4] could be used to filter the data set based on exclusion of items that do not have the desired values of these attributes.

6.2 Query Processing

As mentioned above, the data storage model and search algorithms used in TimeFinder will be made efficient and scalable. Possibilities involve using timeboxes to specify queries over indexed sequence data. For real-valued time-series data, Faloutsos, et al. describe the use of an R*-tree to embed sequences into a multi-dimensional space based on the first few coefficients of the Discrete Fourier Transform of the sequence [10]. After building such an index, we would use timeboxes to specify a query, which would be similarly converted to the frequency domain and searched using the sequential index. This approach would require addressing several details of the transformation used to convert the parameters of timeboxes into the appropriate input for the search algorithm.

Integrating a visual query specification tool based on timeboxes into the general data mining process presents additional challenges. In many cases, the data sets may be too large, and the algorithms too computationally intensive, to work within an interactive system providing frequent, rapid updates of search result displays. In these cases, the interactive query system may be simply one component of a process that combines visualization and exploration with more expensive operations.

Similar concerns regarding retrieval of data from large on-line databases led to the development of query previews [13,25], which were developed to address the difficulty of retrieving information from networked data stores that are too large to be queried directly. Query preview systems pre-compute summary counts of the number of records with various attributes. These smaller summaries are accessed via the network, and a browser is used to restrict the records required, using filtering mechanisms similar to those described above. Once the result set is restricted to a manageably small size, the user can invoke the expensive operation of requesting the specified data.

Variants of this approach might prove useful for using interactive timebox queries in conjunction with expensive data mining operations. For example, a subset of a larger data set (perhaps chosen through some form of statistical sampling) would be placed in an interactive tool similar to TimeFinder. Users would work with this tool to identify patterns of interest, build preliminary understanding of the data set, and develop queries that they would like to run against the whole data set. Once chosen, these queries could be executed against the data set as a whole, using appropriate data mining algorithms.

7. CONCLUSIONS

Research in mining time-series data has led to the development of a wide variety of algorithmic approaches for identifying patterns and extracting useful information. Incorporating these algorithms into systems that support exploration and interactive knowledge discovery is the next step in making these approaches more accessible to a wider range of users and problem domains. A more diverse user population will also stimulate more research, as these users generate questions and problems involving further algorithmic challenges.

Timeboxes support query specification through direct manipulation: a query is created by simply drawing a box on the portion of the graph that corresponds to the range of values of interest. TimeFinder demonstrates the operation of this model on a small data set, while supporting data and query envelopes for displaying overview information and drag-and-drop for query-by-example.

Immediate plans include extension of the timebox model to support increased query expressiveness. Possible extensions include support for queries aimed at finding events at arbitrary time sequences, or with given changes in attribute value. By supporting these queries and others of similar interest to the data mining community, timeboxes become in an interface for data mining applications. This utility will encourage improved algorithms and system design for integrating the interactive search facilities into the computationally intensive, non-real-time processes often associated with data mining applications.

The utility of timeboxes will be a function of the usability of the interface, particularly in comparison with alternative approaches. Empirical studies will help clarify the benefits and drawbacks of timeboxes, while suggesting additional interface improvements.

Finally, we are interested in extending timeboxes to work with different data sets and application domains. Possibilities include the general class of categorical data such as medical histories [19,20], and genetic expression data from microarray experiments [15].

8. ACKNOWLEDGMENTS

Thanks to Martin Wattenberg for providing stock price datasets, and to Erick Baerchke and Hyunmo Kang for valuable feedback. The first author was supported by a fellowship from America Online.

9. REFERENCES

[1] Agrawal, R., Psaila, G., Wimmers, E., and Zaït, M. Querying Shapes of Histories. In Proceedings of 21st VLDB Conference (Zurich Switzerland, September 1995), 502-514.

[2] Agrawal R., and Srikant R. Mining Sequential Patterns. In Proceedings of the Internationall Conference on Data Engineering 1998 (ICDE ‘98) (Taipei Taiwan, March 1995), IEEE Computer Society.

[3] Agrawal, R., Lin, K.I., Sawhney, H.S.., and Shim, K. Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-Series Databases. In Proceedings of the 21st VLDB Conference (Zurich Switzerland, September 1995), 490-501.

[4] Ahlberg, C., and Shneiderman, B. Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. In Proceedings of CHI ’94 (Boston MA, April 1994), ACM Press, 313-317.

[5] Allen, J.F. Maintaining Knowledge about Temporal Intervals. Communications of the ACM, 26(11), November, 1983, 832-843.

[6] Bederson, B.B., Meyer, J., and Good, L. Jazz: an Extensible Zoomable User Interface Graphics Toolkit in Java. In Proceedings of UIST 2000 (San Diego CA, November 2000), ACM Press, 171-180.

[7] Berndt, D.J. and Clifford, J. Finding patterns in time series:A dynamic programming approach. In Advances in Knolwedge Discovery and Data Mining, Fayyad, U.M., Piatesky-Shapiro, G., Smyth, P., and Uthursamy, R., Eds., Menlo Park, CA: AAAI Press, 1996, 229-248.

[8] Carlis, J.V., and Konstan, J.A. Interactive Visualizations of Serial Periodic Data. In Proceedings of UIST 98 (San Francisco CA, November 1998), ACM Press, 29-38.

[9] Card, S.K, Mackinlay, J, D. and Shneiderman, B. Readings in Information Visualization: Using Vision to Think. Morgan-Kaufmann Publishers, San Francisco, CA, 1999.

[10] Faloutsos, C., Ranganathan, M., Manolopoulos, Y. Fast Subsequence Matching in Time-Series Databases. In Proceedings of SIGMOD ’94 (Minneapolis MN, May 1994), ACM Press, 419-429.

[11] Furnas, G. Generalized FisheyeViews. In Proceedings CHI ’86 (Boston MA, April 1986), ACM Press, 16-23.

[12] Gibson, W. MIMSY: A System for Analyzing Time Series Data in the Stock Market Domain. Master’s thesis, University of Wisconsin, Dept. of Computer Science, 1993.

[13] Greene, S., Tanin, E., Plaisant, C., Shneiderman, B., Olsen, L., Major, G., Johns, S. The End of Zero-Hit Queries: Query Previews for NASA's Global Change Master Directory International Journal Digital Libraries 2(2+3), 1999, 79-90.

[14] Hibino, S. and Rudensteiner, E.A. Comparing MMVIS to a Timeline for Temporal Trend Analysis of Video Data. In Proceedings of Advanced Visual Interfaces 1998 (AVI ’98). (May 1998), ACM Press.

[15] Hill, A.A., Hunter, C.P., Tsung, B.T., Tucker-Kellogg, G., and Brown, E.L. Genomic Analysis of Gene Expression in *C. elegans.* Science 290(27), October 27, 2000, 809-812.

[16] Jagadish, H.V., Koudas, N., and Muthukrishnan, S. Mining Deviants in a Time Series Database. In Proceedings of the 25th VLDB Conference (Edinburgh Scotland, 1999), 102-113.

[17] Keogh, E.J., and Pazzani, M. P. Relevance Feedback Retrieval of Time Series Data. In Proceedings SIGIR ’99 (Berkeley, CA, August 1999), ACM Press, 183-190.

[18] Little,J.B., and Rhodes, L. Understanding Wall Street. Liberty Publishing, Inc., Cockeysville, MD, 1978.

[19] Plaisant, C., Milash, B., Rose, A., Widoff, S., and Shneiderman, B. Life Lines:Visualizing Personal Histories. in Proceedings of CHI ’96 (Vancouver, Canada, April 1996), ACM Press, 221-227.

[20] Powsner, Seth M., and Tufte, E.R. Graphical Summary of Patient Status. The Lancet, 344(8919), August 6, 1994, 386-389.

[21] Rafiei, D. and Mendelzon, A. Querying Time Series Data Based on Similarity. IEEE Transactions on Knowledge and Data Engineering, 12(5), September/October 2000, 675-693.

[22] Shneiderman, B., Designing the User Interface. Addison-Wesley, Reading, MA, 1997.

[23] Silva, S.F., and Catarci, T. Visualization of Linear Time-Oriented Data: a Survey. in Proc. of the First International Conference on Web Information Systems Engineering (WISE'00), (Hong Kong, June 2000). IEEE Computer Society.

[24] Spotfire. http://www.spotfire.com. (Accessed February, 2001).

[25] Tanin, E., Plaisant, C., Shneiderman, B. Browsing Large Online Data with Query Previews. In Proceedings of the Symposium on New Paradigms in Information Visualization and Manipulation (NPIVM) 2000 (Washington D.C., November 2000), ACM Press.

[26] Wattenberg, M. Sketching a Graph to Query a Time Series Database. In Proceedings of CHI 2001 (Seattle WA, April 2001), ACM Press.

[27] Yi, B.K., Jagadish, H.V., and Faloutsos, C. Efficient Retrieval of Similar Time Sequences Under Time Warping. In Proceedings of the International Conference On Data Engineering (ICDE '98), IEEE Computer Society Press, 201-208.