Visualization of Linear Time-Oriented Data: a Survey

Sônia Fernandes Silva* and Tiziana Catarci**
*Agorà Telematica Spa, Roma, Italia, e-mail: s.fernandesdasilva@agora.stm.it
**Dipartimento di Informatica e Sistemistica, Università degli Studi di Roma "La Sapienza", Italia
e-mail: catarci@dis.uniroma1.it

Abstract

During the last years databases have been growing in size, varieties of data, number of users, diversity of applications. Many such applications deal with data characterized by the temporal dimension (e.g., medical records, biographical data, financial data, etc.). Typically, end-users of these data are competent in the field of the application but are not computer experts. They need easy-to-use systems able to support them in the task of accessing and manipulating the data contained in the databases. It is well known that visual techniques are suitable in supporting users interacting with large data sets. However, no much research has been carried on specifically on the relationship between visual techniques and time-dependent data. This paper aims at filling this gap by presenting an overview of the main visual techniques for interactive exploration of time-oriented (historical) information.

1. Introduction

The main purpose of adopting a visual representation in a database system is to communicate clearly to the user the information content of the database, concentrating on essential features and omitting unnecessary details [11]. Such information is internally structured in several manners that mainly depend on the data model characteristics, but it must be rendered at the interface level in such a way that any user can easily grasp it. Data characterized by a temporal dimension constitute an essential part for many applications, such as medical and biographical records, multimedia systems, geographic information systems, scientific and statistical applications, etc. In a simplistic view, time can be regarded as just an ordinal dimension in a 2D or 3D visualization, so to exploit one of the several alternatives that were proposed for visualizing linear data (see [52], [53]).

However, visualizing time-related data is more than ordering them along an axis. Indeed, extensions of existing visualization techniques or new techniques have mainly focused on time-oriented interactive exploration and visualization of large datasets. Note that most of them address a particular notion of time (e.g. linear or branching time, discrete or continuous time) in domain-specific applications, since it is difficult to represent the different aspects of temporal data in a single visual structure.

In this paper, we present an overview of the main visual techniques for interactive exploration of time-oriented information. In order to classify temporal data we exploit a framework defined in [22]. Such a framework was originally used to compare and analyze different temporal data models. Furthermore, we characterize time-oriented visualizations as supporting snapshot or slice views, which correspond to the visualization of an instantaneous fact (valid at a single time instant) or an historical fact respectively. The combination of the kind of supported view with the classification provided by the above framework gives rise to the categorization of the visual techniques that we adopt in this paper.

The paper is organized as follows. Section 2 recalls the framework of [22] and introduces the categorization we use in the rest of this paper. Sections 3, 4 and 5 discuss the characteristics of the various visualizations according to such a categorization and presents examples of systems implementing them. Section 6 summarizes the main visual features of the cited systems. Section 7 briefly presents issues related with visual query modalities. Finally, Section 8 draws some conclusions.

2. A Temporal Framework

An object-oriented framework was defined in [22] as a unifying realm for various features found in several temporal data models. Such features gave rise to four design temporal dimensions: *temporal structure*, which provides the underlying ontology and domains for time, *temporal order*, which gives an ordering to time, *temporal representation*, which provides a means to represent time, and *history*, which allows events to be associated with time. According to [22], the *temporal structure* module is comprised of *temporal primitives* supported by a data model, *temporal domains* and *determinacy* over these primitives. The determinacy aspect is out of the scope of this paper.

Temporal primitives can be *absolute* (anchored), where a specific *valid time* (time when the fact is true in the modeled reality [47]) is associated with a fact and is independent of a valid time of another fact (e.g. "March 2000"), or *relative* (unanchored), where the valid time of a fact is related to some other time (e.g. "last week") [47]. There are two absolute primitives, the *instant* (e.g. "January 1995") and the *time interval*, which is a duration of time between two instants (e.g. <"January 1995", "March 2000">). Whereas, the *span* is the relative primitive with a known time duration but no specific starting or ending instant [26]. A temporal domain can be *discrete* or *continuous*. Discrete domains are isomorphic to natural numbers, where a temporal primitive has a successor and predecessor, while continuous domains are isomorphic to real numbers.

A temporal order can be *linear* or *branching*. In a linear order, time flows from past to future in an ordered manner. In a branching order, time is linear from the past to now, when it divides into several futures, each one representing a potential sequence of events.

The linear structure of time may be represented into a hierarchy of *calendars* constituted by specific time intervals expressed with a certain granularity, such as decades, months, weeks, etc. Calendars provide a human interpretation of time [7]. In general, people use *the Gregorian calendar*, which may be expressed in the form <month, day, year>, for scheduling, events planning, etc.

Finally, the *temporal history* is composed of real-world entities which are time ordered. A temporal history may be *valid*, where each entity is associated with a valid time, *transaction*, which associates an entity with a transaction time (the time when a fact is stored in the database [47]), and *event*, which associates an entity with an instantaneous fact, i.e., something occurring at an instant [26]. In this paper we do not consider the transaction history.

As we said in the Introduction, we consider an additional classification criterium for time-oriented visualizations, i.e., if the visualization supports snapshot or slice views. Based on the framework described above and this additional criterium, we propose the following classification:

1. *Slice*, which corresponds to a visualization of valid history, i.e., a visualization of one or more entities (and their attributes and relationships) valid at discrete (continuous) instants or intervals in a *linear* order;
2. *Periodic Slice*, which corresponds to a visualization of valid history at specific discrete (continuous) patterns of time (calendar).
3. *Multi-slice,* which corresponds to a visualization of valid history at discrete (continuous) instants or intervals in a *branching* order.
4. *Snapshot*, which corresponds to a visualization of event history, i.e., a visualization of one or more entities valid at a single discrete (continuous) instant or interval.

In this paper we do not consider multi-slice visualization. Each one of the remaining classes is discussed in one of the following sections.

3. Slice Visualization

Several systems which explore visualization of time-linear data have been proposed in the literature. Most of them, such as [27], [23], [2], [40], [48], [34] propose to visualize temporal data through interactive 2D timelines.

More recent approaches aim at better facilitating the user-interaction through advanced visual techniques, such as visual distortion in 3D visualization, 3D animation, visual metaphors and automatic generation/display of timelines. Furthermore, there are systems that are tailored to specific applications, such as geographical information systems, mobile computing, VR-systems, allowing the visual exploration of *spatio-temporal* data (e.g., [33], [49], [35]).

3.1. Interactive Timelines

A timeline is a graphical or textual display of events in chronological order [34] and is the most used technique for interacting with time-linear visual information. It also allows the user to explore relationships among historical events.

In this section, we present some *interactive timelines*, i.e. timelines that are displayed on a computer screen and may be directly manipulated by the user, so to make easier the chronological navigation of a large temporal data set. Interactive timeline systems proposed in the literature mainly use 2D visualizations, where the time dimension is represented by a slider labeled with discrete periods at specific time granularities (scrollbars are needed when all periods do not fit on the screen).

These interactive timelines are widely used for program debugging and monitoring systems [24], [3], [27], analysis of video data [15], [25], visualization of medical records [55], [42], [30], [14], and personal histories [57], [40], [43]. Most timeline systems are particularly suitable for a specific domain.

An interesting timeline system is *Lifelines* [40], which implements a technique for visualizing individual history in a 2D layout. In *Lifelines* two types of histories are presented: medical patient records and youth criminal histories. The screen is horizontally divided into regions, which are visualized by alternate background colors, where each region represents a specific phase of a person's life. Visual symbols, such as horizontal lines and icons, represent periodic and instantaneous events of a particular entity (person) along a time axis.

Visual cues, such as colors, labels, line thickness and highlighting, indicate the relevance of certain information and relationships between events, as shown in Figure 3.1 (from http://www.cs.umd.edu/hcil/lifelines/). This figure shows a hypothetical medical record overview (see [40]).

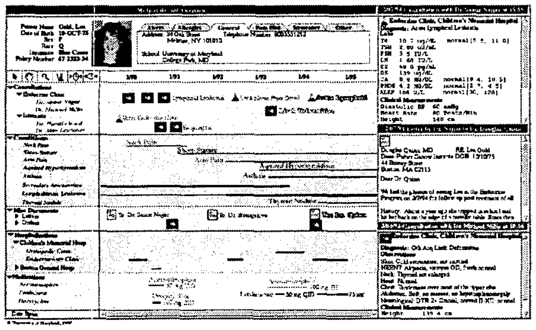


Figure 3.1. ***Lifelines***

Similarly to other systems, Lifelines is thought for a specific domain. It displays small data sets, i.e., only one real-world entity (person) and its attributes are visualized at a time, but it is a rich visualization environment since it provides overview, interactive zooming and filtering (i.e., zoom on a particular period or phase) mechanisms, plus details on demand (i.e., a line or icon can be individually selected). Time axis may be selected at several levels of detail, allowing the view of events at different time granularities (year, month, day).

3.2. Innovative Timelines

Innovative timelines have been proposed to enrich the visualization offered by "traditional" timelines. They make use of distortion techniques [36], [10], modeling features [27], [41], [34], visual metaphors [20], [12], [56], interactive visualization of video and spatio-temporal data [25], [33], [45], [8], [35], in order to better visualize huge amounts of data on the screen and provide the user with more effective mechanisms to explore such data.

The *Perspective Wall* [36] is a well-known distortion technique that makes better use of the available screen space, by integrating detailed and contextual views of large amount of linear information. Mackinlay et al. argue that large data sets are often linearly structured by some metric (e.g. time), but they are not efficiently displayed in a 2D layout (with extensive scrolling), which results in loss of context.

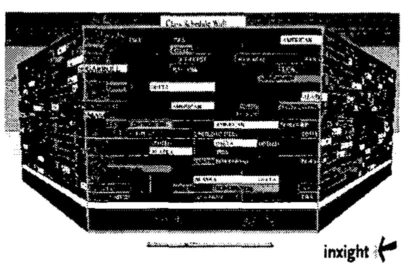


Figure 3.2. ***Perspective Wall***

The solution is distorting such a 2D layout into a 3D visualization, on which the user visualizes the information on a perspective wall, as shown in Figure 3.2, from http://www.parc.xerox.com/istl/projects/uir/images/, where data (contextual information) outside the center area (detailed information) are reduced in size, and represented in two perspective areas, in accordance with the technique known as Fisheye view [21].

The user directly interacts with the wall by selecting an object in one of the three panels. A smooth transition among different views of the wall is pursued. Perspective wall visualizes several entities at a time but has the limitation of visualizing only one entity attribute.

Approaches exist which allow the automatic generation of domain-dependent [27], [48] and domain-independent [41], [34] timelines. The work described in [41] proposes an information architecture for generating timelines of personal histories in several application domains. The work described in [34] deals with the logical modeling of timelines for visualizing metadata.

In [34] a two-layer, namely content and display layers, object-oriented model is presented for creating generalized and domain-independent timelines, with a large diversity of temporal entities and many types of inter-relationships among them. Entities and relationships, as well as their visual properties and cues, such as shape, color intensity, size, dimension, pixel distance, etc., may be interactively generated by the end-user. For example, Figure 3.3 (from http://www.csdl.tamu.edu/~vijayk/timelines.html) illustrates a timeline of Western music composers by using the *tmViewer* tool, a prototype implementation of the ITER model for visualizing temporal relationships among historical events.

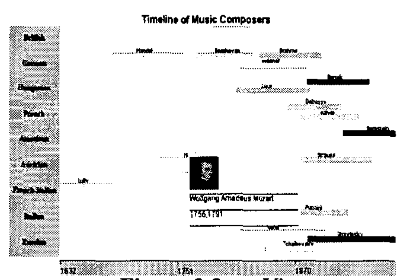


Figure 3.3. ***tmViewer***

Visual metaphors have been explored for a better visual perception of the linear evolution of data. For instance, *Lifestreams* [20] proposes an enhancement of the desktop metaphor, where a time-ordered stream of documents is presented to replace conventional files and directories. Other approaches, e.g., [12] and [56], make use of visual metaphors for analyzing the evolution of the information on the Web.

PeopleGarden [56] exploits some visual metaphors for exploring not only personal histories but also the history of the interactions which take place among them in any on-line environment (e.g. web-based messages). It uses a *flower metaphor* for creating individual data portraits, and a *garden metaphor*, a 2D layout containing several visual flowers vertically arranged, for visualizing patterns of interaction among a group of users during a specific time period (see Figure 3.4, from http://graphics.lcs.mit.edu/~edu/becca/pgarden/). Each user is represented as a flower and its messages are rendered as petals of the flower, arranged by time in a clockwise fashion. The flower height denotes the period in which a user has been at the board.

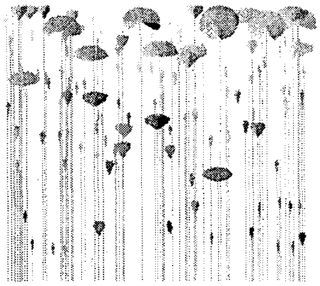


Figure 3.4. ***PeopleGarden***

A new visualization, called *TimeTube*, which integrates visual metaphors and 3D visualization, was introduced in [12]. Time Tube organizes and visualizes the evolution of Web sites over specific time periods. More precisely, it is used to aid Web producers or users to analyze some issues involving the evolution of *Web ecologies*, in order to address queries (see [12]) about content (e.g. Which information has been modified?) and usage (e.g. Which is the least popular information?) of Web information and their inter-relationships (e.g. When does the new information become popular?) along time. Time Tubes are composed of one or more *Disk Trees*, which represent the hyperlink structure of a Web site. For example, Figure 3.5 (from http://www.parc.xerox.com/istl/projects/images/) illustrates a Time Tube composed of four Disk-Trees, representing the evolution of the site www.xerox.com during April 1997. Each Disk Tree represents one week of April, and the third dimension is the time.

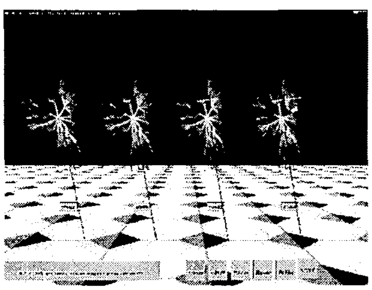


Figure 3.5. ***TimeTube***

3.3. Visualization of Spatio-Temporal Data

There are systems, such as geographical information systems, mobile computing, virtual environments, multimedia systems, etc, which allow the interactive visual exploration of video and spatio-temporal data. In this section we recall significant examples from [25], [33] and [35].

MMVIS is a new approach which extends the VIS (Visual Information Seeking) technology [5] for the analysis of temporal trends in video data. It manipulates spatio-temporal characteristics of multimedia data by integrating a temporal visual query language, TVQL [25], and a temporal visualization of results in an iconic representation, where icons denote the different types of events in the video dataset. Visual symbols and cues (e.g. highlighting, geometric symbols, transparent overlays, bars) are used for visualizing temporal occurrences of these events and relationships between them. The visualization may be dynamically changed through the interactive manipulation of the TVQL query filters. TVQL is described in Section 6.

The work presented in [33] adopts advanced visual techniques (infinite zoom, animation, semitransparency) within a 3D landscape. More specifically, it presents a domain-dependent dynamic timeline in a 3D timespace scenario for interactive exploration of photographers' histories. First, a contextual view of information is spread on a virtual scenario. In order to get the detailed view of a piece of information, the user interactively zooms-in over the individual timelines of the photographer. As the user moves closer to the individual timeline more detailed information appears; as the user moves away this information becomes transparent. This is realized through an animated visual transition, by providing an interactive "time-travel" in the environment.

Finally, *VRML History* is an interesting proposal of incorporating the time dimension into VRML. VRML History allows for visualizing spatio-temporal data as multimedia objects and interactively exploring them in a virtual environment. Such an environment provides the user not only with real time 3D animation, but also with the possibility of accessing past and future 3D animations of the objects, which are simulated in specific time periods. For example, Figure 3.6 (from http://www-winfo.unisegen.de/vrmlHistory/examples/index.html) illustrates a screenshot which contain a historical view of a

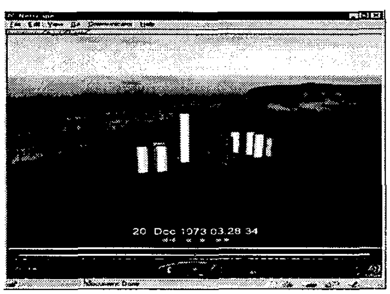


Figure 3.6. ***VRML History***

city as result of a temporal query to a temporal GIS-database. This approach overcomes the limitations found in video animation, which does not support 3D navigation, and VRML-files, which does not control the time dimension during animated presentations (e.g. animation stops or goes backward and forward with different speeds).

4. Periodic Slice Visualization

Visual techniques have been proposed explicitly dealing with the visualization of periodic patterns in historical data. For instance, a proposal for personal scheduling is a visual calendar which represents an enhancement of traditional paper calendars. The spiral visualization and other visual techniques have been adopted for visualizing time-series data.

4.1. Visual Calendars

Some visualization techniques give emphasis to specific periodic patterns, mainly used in scheduling applications, such as the *electronic calendars*. Electronic calendars [31], [6], [50], [37], [39], have arisen as enhancements of paper calendars, since they exploit several visual clues, such as highlighted view of related events, graphical symbols as patterns for cyclic events, etc.

In [6], a calendar scheduling system for maintaining personal calendars and scheduling meetings is presented, called *Visual Scheduler*. A more recent visual calendar is the *Calendar Visualizer* [37], which is also used for personal and meeting schedules, and exploits advanced graphical technology such as 3D visualization and interactive animation. The Calendar Visualizer [37] comprises two visualizers that enhance the exploration of personal and group scheduling events. The *Spiral Calendar* is used for accessing an individual daily schedule in a 3D spiral and the *Time Lattice* for analyzing the temporal relationships among the schedules of groups. Figure 4.1 (from [37]) illustrates the Spiral Calendar. Note that different calendars are visible at multiple levels of detail (depth) and are connected by transparent pyramids, similarly to a spiral lap. The user may focus on a detail of any calendar (e.g. week) by clicking on it and the selected calendar becomes the closest one on the spiral. The calendars with thinner time granularities (e.g., day) are not visualized. Information detail and context are tightly integrated through the 3D spiral.

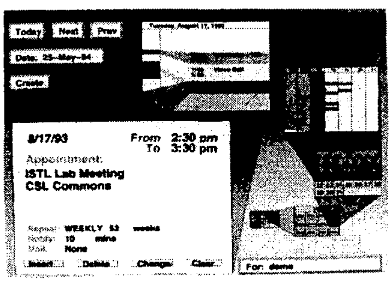


Figure 4.1. ***Calendar Visualizer***

Another approach which integrates the scheduling and personal information management, by presenting the timeline and calendar views is described in [43]. In this paper, a prototype called *TimeScape* is presented, which implements the *Time-Machine Computing* concept [44], where the user may access past and future desktop states of personal information in a desktop environment. Such states may be visualized in three different ways: snapshot (described in Section 5), timeline and calendar views (the interconnection among views may be also visualized).
Figure 4.2 illustrates the timeline view, from http://www.csl.sony.co.jp/person/rekimoto.html. In the timeline view, past and future desktops are visualized as horizontal lines. The past (future) lines are positioned on the left (right) of a central desktop, which represents the current one (zooming mechanism is used for changing the time granularity of the timeline). In the calendar view, items which compose a desktop are displayed as date cells in a simple calendar. Only one calendar may be seen at a time.

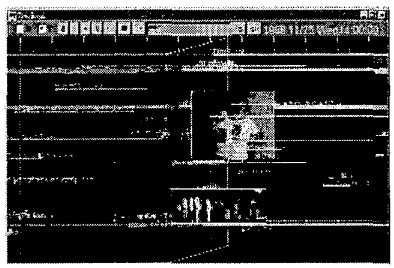


Figure 4.2. ***Timeline View***

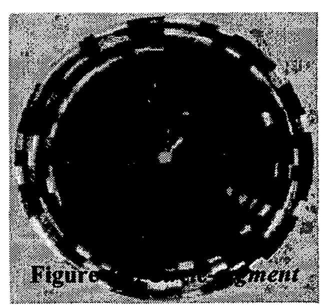
The user can relate these views to a specific time.

4.2. Visualization of Time-Series Data

Visual techniques exist which explore periodic patterns found in data mining, where time is considered a dimension in a multidimensional dataset (time-series data) and the change of data values is tightly associated with a specific pattern of time [16]. Some visual techniques (e.g., [9]) have been created ad-hoc for visualizing periodic patterns in large datasets, whereas other ones are more general (e.g., [29]).

As for the latter, in [28] some visualization techniques are described for the effective exploration of multidimensional data [28], e.g., pixel-oriented visualization techniques which may be applied on time-series data.

For example, Figure 4.3 (from [29]) illustrates a circle-segment technique which visualizes k-dimensional data



(resulting from the execution of a specific query) as a circle divided into k segments. Such data may be time-ordered. Other visual techniques, such as the spiral space-filling curve may be used for serial data.

The approach presented in [9] is specifically thought for visualizing periodic patterns of serial (continuous) data. It uses a planar spiral as visualization technique for exploring serial periodic data (defined in [9] as data which have a continuous dimension that exhibits periodicity). Serial data are explored along the spiral and periodic data are explored along the radii. For example, Figure 4.4 (from [9]) illustrate the spiral visualization used in an application domain. The lap and spoke of the spiral represent a specific year and month respectively (see [9] for more details). Using the spiral visualization technique it is possible to simultaneously identify serial and periodic variations of a single datum or a large amount of data.

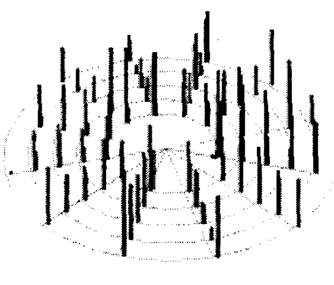


Figure 4.4. *Spiral Visualization*

5. Snapshot Visualization

Most of the visual techniques we have recalled in the above sections provide the user with the interactive exploration of information across several time periods in a linear or branching order, to identify temporal trends/evolution of one or more objects. However, in some applications, such as personal histories, image sequences, etc, an additional functionality is needed that allows the snapshot visualization of an information, i.e., visualization of data valid at a single discrete (continuous) instant or interval. The snapshot visualization is very useful in presenting current and past instantaneous facts, highlighting a facet of a current view or a past state of an information.

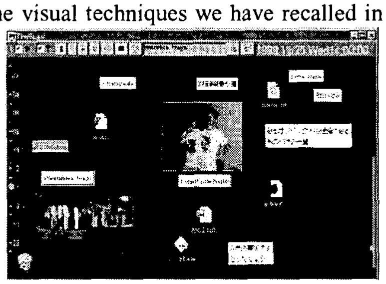


Figure 5.1. *TimeScape*

Some timeline-based systems, such as *Lifelines* [40] and *TimeScape* [43] support the snapshot visualization. As we said in Section 4.1, the TimeScape system visualizes past and future desktop states of personal information in a desktop environment in three different ways: snapshot, timeline and calendar views.

The snapshot view visualizes the desktop items (file and application icons, etc) at a time instant. The instant is visually represented as a "time-travel" dial, successor and predecessor instants may be accessed by navigation buttons, as shown in Figure 5.1 (from http://www.csl.sony.co.jp/person/rekimoto.html). The system restores the computer state at a specific instant, by allowing a "time-traveling" interaction.

In geographical information systems, dynamic landscapes may represent a discrete sequence of images (e.g. vegetation covers) generated by a particular event (e.g. fire disturbance), as those found in *SELES* [18], a spatially explicit landscape event simulator, or in *VRML History* [35] (described in Section 3.2). SELES [18] allows one to perform the visual analysis of a sequence of landscape event information in order to model future states of forested landscapes for planning and safeguarding. A further work [10] combines the sequence of images in a 3D cube (Figure 5.2 from [10]), with spatial (left hand side) and temporal (right hand side) layers. The spatial layer represents vegetation cover at a certain time instant. Note that each grid cell indicates a dominant vegetation type of the area it represents. The temporal layer towards the front represents sequences of changes of the vegetation type. A visual distortion technique is applied to the 3D cube in order to facilitate the exploration of the temporal information.

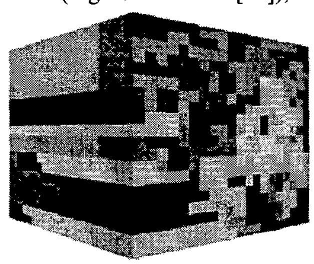


Figure 5.2. *SELES*

6. System Features

Table 1 and 2 summarize the main visualization and interaction features the systems we surveyed in the previous sections exhibit. The column headers have the following meaning: *Periodic Pattern* and *Snapshot View* have to be intended as described in the previous sections; *Multiple Calendars* is a visual technique which simultaneously presents two or more calendars on a single data set; *Complex Entity* stands for the ability to visualize an entity and its properties; *Entity relations* means the availability of a visual technique which provides one with visual cues/parameters to highlight relationships (causal, similarity, temporal, hierarchical, etc) among several entities; *User-Defined Display* means that the user may change/define the display; *Pattern/trends* means that general patterns (not periodic) may be visually analyzed;

Focus + Context stands for the possibility of simultaneously visualizing general and detailed information; *General* means not domain-dependent.

Table 1. Visualization Features

	Periodic Pattern	Snapshot View	Multiple calendars	Complex Entity	Entity relations	User-defined display	Patterns/ Trends	Focus + Context	General
Visual Scheduler									
Perspective Wall									
Visual Calendar/ Time Lattice									
Dynamic Timelines									
Lifestreams									
Lifelines									
MMVIS									
SELES									
ITER/ TmViewer									
Spiral Visualization									
TimeTube									
PeopleGarden									
TimeScape									
VRML History									

Table 2. Interaction features

	Overview	3D navigation	Time navigation	Zooming	Filtering	Temporal Filtering	Details on demand
Perspective Wall							
Visual Calendar/ Time Lattice							
Dynamic Timelines							
Lifestreams							
Lifelines							
MMVIS							
SELES							
ITER – TmViewer							
Spiral Visualization							
TimeTube							
TimeScape							
TVQE							
VRML History							

As for Table 2, *Time navigation* means the availability of a specific time-traveling mechanism, with which the user may restore the data visualization to a specific time; *Overview* is the visual technique providing an overview of the entire data collection; *Zooming* allows one to zoom on items of interest, *Filter* filters out uninteresting items, *Details on demand* means that it is possible to selects an item or a group of items and get details when needed [46].

7. Temporal Visual Queries

In the previous sections various techniques for visualizing time-related information have been classified and discussed. Typically, almost all cited systems provide the user with interaction mechanisms to analyze the visualized data. However, very often such mechanisms just allow the user to browse and visually manipulate the data, they do not support querying capabilities for extracting further information. Sometimes what is visualized is the result of a previous query, which was expressed interacting with a different visualization of the data. This is also due to the fact that traditionally what is visualized for the query purpose is the schema of the database, while the actual database instances constitute the query result to be displayed to the user. In this section we briefly present examples of systems combining a query phase with a result visualization phase, each phase exploiting its own visualization and interaction mechanisms.

Severals temporal data models and corresponding textual query languages have been proposed in the literature, where special clauses and predicates are added to the original language in order to deal with temporal aspects (significant references can be found in [13], [38], [54]). These languages retain the usability problems of the originating query languages, such as the intrinsic syntactical complexity and the lack of a global view of the data of interest together with their interrelationships. Considering specifically the temporal relational languages, the user must be familiar with concepts such as tuple, attribute time-stamping and temporal joins, as well as sintax and semantics of temporal predicates.
However, there is a limited amount of research concerning the investigation of visual interfaces for querying temporal databases. Notable proposals are [32], ERT/vql [51], TVQL [25], TVQE [19].

The visual query language of Kouramajian and Gertz [32] comprises some visual constructors which allow the formulation of temporal queries, based on a temporal extended Entity-Relationship model, but it does not explicitally visualizes temporal entities and relationships and the database schema is represented as an E-R diagram. Such a model-based representation may be cumbersome to non-expert users. A similar approach is followed by the visual language for the ERT model, ERT/vql [51]. This system exhibits a result visualization phase where advanced interactive visualization techniques are applied over the query result. However, during the query construction phase, the user needs to adopt a textual syntax and a model-based visual representation of the schema.

As we said in Section 3.3, MMVIS [25] integrates a temporal visual query language (TVQL) with a temporal visualization of results. TVQL contains dynamic query [5] filters, i.e, the temporal visualization is dynamically updated as temporal query filters are adjusted. Through the query filters, it is possible to specify the primitive temporal relationships between events defined by Allen [1]. In [25], new index data structures and query processing strategies were developed for handling TVQL queries.

TVQE [19] is a visual query system which provides the user with a friendly environment to interact with temporal databases. The system adopts a diagrammatic representation of the database schema and a "graphical notebook" as interaction metaphor. The TVQE interface contains two panels, namely Schema and Interaction Windows. The Schema window is a panel on which the database schema is visually represented as a top-down tree, as a graph (database schema) and as a

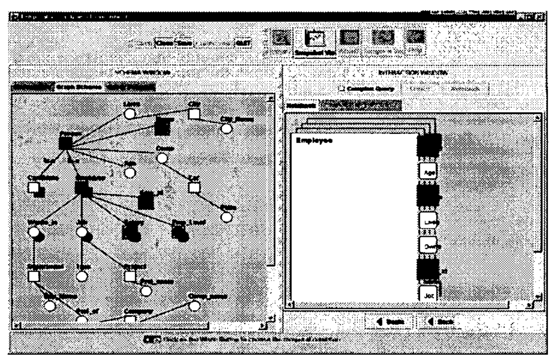


Figure 7.1. ***TVQE Environment***

subgraph (database subschema), where the three representations share the same panel. Through the Schema window, the user has a global view of the classes of the schema and their interrelationships. The Interaction window is a panel on which a database schema is visually displayed as a "graphical notebook". The user interacts with the graphical notebook by selecting its indices, as shown in Figure 6.1.The two windows are synchronized: for each selected index, the corresponding node in the Schema window is also selected.

Nodes in the panel in the Schema window may assume three states, unselected for the query, selected for the query and displayed for the query result. The user can continuously change the status of the node, by clicking on the corresponding index. Next, the system displays the sub-schema.. Through the use of dialog boxes, the user specifies the query condition.

Usability tests showed that the specification of some queries by searching indices in a graphical notebook was enjoyed by the users more than the direct manipulation on the diagrammatic representation, since s/he prefers to interact with something more familiar. However, the visual representation of a schema as a notebook is in some sense less rich than the diagrammatic one, since a diagram favor the visualization of relationships between concepts. So, the system integrates the diagrammatic and iconic representations, emphasizing the usage of the iconic representation as the interaction media. The result of the user's query is visualized by using the *dynamic query* approach introduced in [4].

8. Conclusions

The temporal dimension is becoming more and more important in modern database applications, which are spread over the world and need to be accessed by several people with different expertise and needs. This is driving the development of ad-hoc visualizations and interaction mechanisms which address the specific requirements of time-dependent information. It is now recognized that the initial approaches, just considering the time as an ordinal dimension in a 2D or 3D visualizations, are inadequate to capture the many characteristics of time-dependent information. More sophisticate and effective proposals have been recently presented.

However, none of them aims at providing the user with a complete framework for visually managing temporal-related information. Also, very often systems exhibiting significant visualizations lack in the interaction mechanisms, which do not allow the user to powerfully manipulate the data. Trying to overcome this limitation, systems have been proposed in which the extraction of information is separated from the visualization. However, this approach forces the user to continuously switch between two different environments, possibly causing her/him confusion and cognitive overhead. In summary, much research is still needed to come up with a satisfactory visual environment where the user could effectively interact with time-dependent information.

References

[1] J.F.Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, Vol. 24, N.11, pp. 832-845, 1983.

[2] R.B.Allen. Interactive Timelines as Information System Interfaces. In *Symposium on Digital Libraries, Japan*, pp. 175-180, 1995.

[3] M.Abrams, N.Doraswamy, A.Mathur. Chitra: Visual Analysis of Parallel and Distributed Programs in the Event, Time, and Frequency Domains. *IEEE Transaction on Parallel and Distributed Systems*, Vol. 3, N. 6, pp. 672-685, 1992.

[4] C.Ahlberg, C.Williamson, B.Shneiderman. Dynamic Queries for Information Exploration: an Implementation and Evaluation. In *Proc. of ACM CHI'92*, pp. 619-626, 1992.

[5] C.Ahlberg, B.Shneiderman. Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. In *Proc. of ACM CHI'94*, pp. 619-626, 1994.

[6] D.Beard et al. A Visual Calendar for Scheduling Group Meetings. In *Proc. of ACM CSCW'90*, pp 279-290, 1990.

[7] C.Bettini et al. A Glossary of Time Granularity Concepts, In *Temporal Databases: Research and Practice*, Etzion et al. (Eds), Springer, pp. 406-413, 1998.

[8] I.M.Brown. A 3D User Interface For Visualization of Web-Based Data-Sets. In *Proc. of ACM GIS'98*, Washington DC, USA, 1998.

[9] J.V.Carlis, J.A.Konstan. Interactive Visualization of Serial Periodic Data. In *Proc. of ACM UIST'98, San Francisco, CA*, pp. 29-38, 1998.

[10] T.Carpendale, M. Sheelagh, A.Fall, D.J.Cowperthwaite, J.Fall, F.D.Fracchia Visual Access for Landscape Event based Temporal Data. In *Proc. of Visualization '96*, p. 425, 1996.

[11] T.Catarci, M.F.Costabile, S.Levialdi, C.Batini. Visual Query Systems for Databases: A Survey. *Journal of Visual Languages and Computing*, Vol. 8, N. 2, pp. 215-260, 1997.

[12] E.H.Chi et al. Visualizing the Evolution of Web Ecologies. In *Proc. of ACM CHI'98, Los Angeles CA USA*, pp. 400-407, 1998.

[13] J.Chomicki. Temporal Query Languages: A Survey. In *Proc. 1st Int. Conf. on Temporal Logic*, Springer-Verlag, pp. 506-534, 1994.

[14] Y.Shahar, C. Combi. Timing is Everything: Time-Oriented Clinical Information Systems. *Western Journal of Medicine*, Vol. 168, pp. 105-113, 1998.

[15] M.Davis. Media Streams: An Iconic Language for Video Annotation. *Telektronik 4.93:Cyberspace*, Vol. 89, N. 4, pp 59-71, 1993.

[16] R.Elmasri, J.Y.Lee. Implementation Options for Time-Series Data. In *Temporal Databases: Research and Practice*, Etzion et al. (Eds), Springer, pp. 115-128, 1998.

[17] M.Erwig, R.H.Gutting, M.Schneider, M.Vazirgiannis. Spatial-temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases. In *Proc. CHOROCHRONOS Workshop on Spatio-Temporal Database Systems*, Austria, 1997.

[18] J.Fall, A.Fall Seles: A Spatially Explicit Landscape Event Simulator. In *Proc. GIS/Environmental Modeling Conference*, pp. 104-112, 1996.

[19] S.Fernandes Silva, T.Catarci. Graphical Interaction with Historical Databases. In *Proc. 11th Int. Conf. on Scientific and Statistical database Management (SSDBM'99)*, 1999.

[20] E.Freeman, D.Gelernter. Lifestreams: A Storage Model for Personal Data. *ACM Sigmod Bulletin*, March 1996.

[21] G.W.Furnas. Generalized Fisheye Views. In *Proc. of ACM CHI'86, Boston MA, USA*, pp. 16-23, 1986.

[22] I.A.Goralwalla, M.T.Ozsu, D.Szafron. An Object-Oriented Framework for Temporal Data Models. In *Temporal*

Databases: Research and Practice, Etzion et al. (Eds), Springer, pp. 1-35, 1998.
[23] B.L.Harrison, R.Owen, R.M.Baecker. Timelines: An Interactive System for the Collection of Visualization of Temporal Data. In *Proc. of Graphics Interface 94*. Canadian Info. Processing Society, 1994.
[24] M.T.Heath, J.A. Etheridge. Visualizing the Performance of Parallel Programs. *IEEE Software*, Vol.8, No.5, pp. 29-39, 1991.
[25] S.Hibino, E.A.Rundensteiner. MMVIS: Design and Implementation of a Multimedia Visual Information Seeking Environment. In *Proc. of ACM Multimedia 96*, Boston MA, USA, pp. 75-86, 1996.
[26] C.S.Jensen et al. The Consensus Glossary of Temporal Database Concepts. In *Temporal Databases: Research and Practice*, Etzion et al. (Eds), Springer, pp. 367-405, 1998.
[27] G.M.Karam. Visualization Using Timelines. In *Proc. of Intl Symposium on Software Testing and Analysis (ISSTA)*, 1994, also in SIGSOFT, *ACM Software Engineering Notes*, 1994.
[28] D.Keim. Pixel-Oriented Database Visualizations. *SIGMOD Record*, Special Issue on Information Visualization, pp. 35-39, 1996.
[29] D.Keim. Visual Techniques for Exploring Databases, Invited Tutorial. In *Proc. Int. Conf. on Knowledge Discovery in Databases* (KDD'97), Newport Beach, CA, 1997.
[30] D.Kilman, D.Forslund. An International Collaboratory based on Virtual Patient Records, *Communications of the ACM*, Vol. 40, N.8, pp. 111-117, 1997.
[31] C.M.Kincade, P.B.Dupont, A.R.Kaye. Electronic Calendars in the office: An Assessment of User Needs and Current Tecnhology. *ACM Transactions on Office Information Systems*, Vol. 3, N. 1, pp. 889-102, 1985.
[32] V. Kouramajian, M.Gertz. A Visual Query Editor for Object-Oriented Databases. In *Proc 14th Int. Conf. on O-O and E-R Modeling*, pp. 388-399, 1995.
[33] R.L.Kullberg. Dynamic Timelines: Visualizing the History of Photography. In *Proc. of ACM CHI'96*, pp. 386-387, 1996.
[34] V.Kumar, R.Furuta, R.B.Allen. Metadata Visualization for Digital Libraries: Interactive Timeline Editing and Review. In *Proc. of ACM Digital Libraries, Pittsburgh, USA*, pp. 126-133, 1998.
[35] H.Luttermann, M. Grauer. VRML History: Storing and Browsing Temporal 3D-Worlds. In *Proc. of ACM VRML'99, Paderborn Germany*, pp. 153-160, 1999.
[36] J.D.Mackinlay, G.G.Robertson, S.K.Card. The Perspective Wall: Detail and Context Smoothly Integrated. In *Proc. of ACM CHI'91, New York*, 1991, pp. 173-179.
[37] J.D.Mackinlay, G.G.Robertson, R.DeLine. Developing Calendar Visualizers for the Information Visualizer. In *Proc. of ACM UIST'94*, pp. 109-118, 1994.
[38] G.Ozsoyoglu, R.Snodgrass. Temporal and Real-time Databases: A Survey. *IEEE TKDE*, Vol. 4, N. 7, pp. 513-532, 1995.
[39] L.Palen. Social, Individual and Technological Issues for Groupware Calendar Systems. In *Proc. of ACM CHI'99*, pp. 17-24, 1999.
[40] C.Plaisant, B.Milash, A.Rose, S.Widoff, B.Shneiderman. Lifelines: Visualizing Personal Histories. In *Proc. of ACM CHI'96, Vancouver, Canada*, April, 1996, pp 221-227.
[41] C.Plaisant, B. Shneiderman, R. Muhslin. An Information Architecture to Support the Visualization of Personal Histories. Information *Processing & Management*, Vol. 34, N. 5, pp. 581-597, 1998.
[42] S.M.Powsner, E.R.Tufte. Graphical Summary of Patient Status. *The Lancet*, Vol. 344, N. 8919, pp. 386-389, 1994.
[43] J.Rekimoto. TimeScape: A Time-Machine for the Desktop Environment. In *Proc. of ACM CHI'99 Extended Abstracts*, pp. 180-181, 1999.
[44] J.Rekimoto. Time Machine Computing: A Time-centric Approach for the Information Environment. In *Proc of ACM UIST'99, Asheville, NC*, pp. 45-54, 1999.
[45] J.N.Robinson. Visualize a Port in Africa. In *Proc. of the 1997 Winter Simulation Conference*, pp. 70-77, 1997.
[46] B.Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. http://www.cs.umd.edu/projects/hcil/ CS-TR-3655 ISR-TR-96-66, July 1996.
[47] R.T.Snodgrass. Temporal Databases. In *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, Springer-Verlag, LNCS 639, pp. 22-64, 1992.
[48] Tom Snyder Productions. Timeliner 4.0, *Tom Snyder Productions*, 1997.
[49] K.Taylor, M.Cameron, J.Haines. An Integrated Informatin System on the Web for Catchment Management. In *Proc. of ACM GIS'98, Washington DC, USA*, pp. 8-13, 1998.
[50] F.N.Tessler. Desktop Calendars. *Macworld* 10(7):104-109, 1993.
[51] B.Theodoulidis et al. Interactive Querying and Visualization in Temporal databases. In *Proc. Reasoning Workshop of the 4th DOOD Conf.*, 1995.
[52] E.R.Tufte. *The Visual Display of Quantitative Information*. Graphic Press, Cheshire, Connecticut, 1983.
[53] E.R.Tufte. *Envisioning Information*. Graphic Press, Cheshire, Connecticut, 1990.
[54] Y.Wu, S.Jajodia, X.Wang. *Temporal Database Bibliography Update. Temporal Databases: Research and Practice*. Springer-Verlag, Berlin, 1998.
[55] J.C.Wyatt. Clinical Data Systems, Part 1: Data and Medical Records. *The Lancet*, 344, pp. 1543-1547, 1994.
[56] R.Xiong, J.Donath. PeopleGarden: Creating Data Portraits for Users. In *Proc of ACM UIST'99*, Asheville, NC, pp. 37-44, 1999.
[57] Xsoft Corp. Visual Recall: Personal Edition, version 2.0, 1995.