Using Interactive, Temporal Visualizations for WWW-based Presentation and Exploration of Spatio-Temporal Data

Hartmut Luttermann and Manfred Grauer

University of Siegen, D-57068 Siegen, Germany
{hartmut,grauer}@fb5.uni-siegen.de
http://www-winfo.uni-siegen.de/winfo/english

Abstract. In recent years, spatio-temporal data are increasingly recorded in database-enabled information systems. To share the benefit, often, a public data access is provided utilizing WWW-based visual interfaces. If spatio-temporal data are queried, data visualization and exploration in a spatial and temporal context is needed to better present and analyze spatio-temporal behaviours and relationships. In existing WWW-based interfaces of temporal geographical information or scientific data visualization systems, only videos, static visualizations (images) or, recently, VRML-worlds (Virtual Reality Modeling Language) are applied which support neither 3D- nor temporal navigation, respectively. Additionally, these interfaces offer only small support for further exploratory tools according to the temporal domain and to conceptual database entities which users are often aware of. In this paper, we propose an extension to VRML which supports the description of spatio-temporal (temporal 3D-)graphics and entities by integrating the valid time dimension and the entity concept. This is realized by a set of new VRML-nodes representing temporal geometries, time references and database entities and a set of interactive visualization techniques extending standard VRML-browsers to perform temporal navigation and to further explore the visualization during presentation. One result of this proposal is that a new type of visualizations, i.e. interactive, temporal visualizations (animations), can be generated, stored in files, integrated in Web-pages and explored spatially and temporally by interaction or animation. This is demonstrated by two decision support applications from water management and urban planning.

1 Introduction

In recent years, the amount of spatio-temporal data stored in conventional file systems or in modern GIS- and data warehouse applications has increased dramatically. In many applications these data are displayed and analyzed to obtain knowledge on technical and environmental processes as well as to plan, simulate, explain and evaluate management decisions.

In order to analyze spatio-temporal data, data visualization techniques are used [3, 34]. This involves several steps: Querying and filtering the data set from the database or file system by applying a database query language (e.g. SQL) or a visual query tool, mapping the data into some graphics by applying specific data type- and task-dependent visualization methods, and then visually analyzing the graphics in relation to spatio-temporal behaviours and relationships of conceptual entities as well as certain inherent structures of linear or cyclic processes. These steps are done interactively and iteratively to extract new knowledge or to create visualizations for explanation and public presentation.

Often, public access to the data sets of spatio-temporal databases is provided to facilitate data commercialism or cooperative work. In this case, the main graphical user interface between the "everyone" user (e.g. practioneers, decision makers, researchers) and databases or database applications (e.g. data archives, e-commerce) is becoming WWW-pages. This is the platform most preferred because of its simplicity, ease of use, system-independency and the possibility to integrate and combine different data types in one interface.

In existing WWW-based visual interfaces of temporal geographic information systems and scientific data visualization servers only static time-referenced visualizations (images and, recently, static VRML-worlds) [11, 20, 38, 2] and, in order to keep the temporal nature of the data, videos [16, 20] or, recently, animated VRML-worlds [6, 38] are used to visualize spatio-temporal data. Although video-players offer controls for temporal navigation of the video (e.g. pause, backward), they do not support a 2D/3D-navigation in the video-content. On the other side, VRML itself permits 3D-navigation, but does not feature a technique which controls the time domain of 3D-animations during the presentation.

However, with data visualizations more simple and easy to use, exploratory tools are wanted by the end user to facilitate private analysis and learning [6, 26, 18]. These tools change the way the data are viewed and may help different features (information aspects) of the data to be made salient [41]. Besides exploring the spatial and temporal dimension of the data (with techniques like temporal navigation, spatial and temporal focusing and brushing, temporal comparing [16, 27]) the conceptual data structures (entity modeling) are becoming more important for the viewer working with data sets of databases. This makes exploratory tools available which link the entity description (i.e. the non-spatial attributes) with the graphic (e.g. linked brushing).

In this paper, we propose a seamless extension of VRML which supports the description of spatio-temporal graphical objects and entities by conceptually integrating a valid time dimension and an entity concept. VRML is a standardized file format and high-level description language for interactive 3D-graphics on the internet which is supported by many browsers. It integrates user interaction with, an event/message passing between the objects in the visualization and a prototying and programming mechanism to extend the language with new object types and behaviour [42]. This proposal is realized by a set of new VRML-nodes expanding VRML to be a description language for temporal 3D-graphics with entity features, and by a set of interactive visualization techniques extending standard VRML-browsers to perform temporal navigation, object identification and the exploration of temporal data and entity characteristics. With this extension, visualizations of spatio-temporal data sets and entities can be described as *interactive (explorable), temporal visualizations (animations)*, stored in VRML-files on the internet, integrated in Web-pages, and explored in a spatial, temporal, and thematical context in standard VRML-browsers during presentation possibly being a part of a public WWW-based visual database interface.

The paper is organized as following: In section 2, we review existing visualization techniques for spatio-temporal data, and in section 3, we give a short overview on current graphic technologies competing with VRML. In section 4, we show which different features of spatio-temporal data exist possibly inherent in visualizations and in section 5 which types of exploratory tools are needed to reveal it. In section 6, we explain the newly designed VRML-nodes and their implications. In section 7, we present some example applications and conclude this paper with a discussion and the plans for future works in section 8.

2 Visualizing Spatio-Temporal Data

In cartographic applications (e.g. cadastral systems), the visualization of spatio-temporal data mainly comprises the mapping of states and changes on static maps [22]. Whereas time-referenced states are visualized in 2D or 3D applying traditional cartographic visualization methods (e.g. choropleth maps, isarithmic maps, dot maps, net maps), change is mapped on temporal maps classified in static and strip maps. Strip maps are multiple time-referenced snapshots ("small multiples", cartoon) where change is shown via the differences between the individual maps depicting subsequent snapshots. In static maps, time is mapped on a visual design variable [4] (i.e. shape, size, color, orientation, location, lightness or texture) resulting in e.g. space-time-objects or time series plots (time as a geometric dimension), particle trace maps (spatial behaviour of particles), super-imposed (overlayed) maps/spaces, arrow/flow maps (symbolized motion), isochrone diagrams, period-referenced difference maps/spaces.

However, with static maps it is difficult to view the data both in a spatial and temporal context facilitating an understanding of processes rather than states and the investigation of spatio-temporal relations and patterns. Generally, on dynamic presentation systems (and devices) time is used as a new visual design variable inducing not only one but six new dimensions [25]: duration, frequency, ordering, rate of change, synchronization and presentation time. However, presentation time is the most used dimension to present spatio-temporal data resulting in the generation of animations and videos.

Two different types of animations are distinguished [9, 31]: a) *temporal animations*: a change in the presented object's location, shape or attributes is based on temporal data, i.e. there is a direct relation between world time which is part of the data and presentation (system/real) time which is used to show the animation; and b) *cartographic animations*: the animation method is applied to non-temporal data, such as the camera position (viewing), light (rendering) or other animation variables and techniques [33], e.g. object change/motion (e.g. bubbles in water showing the stream), time-variant symbols for data values (e.g.

rain symbol), presentation change (e.g. highlighting). Also, user interaction may be a cause for cartographic animations [31], e.g. observer motion (e.g. zooming), change of visualization parameters (e.g. isolines, color-mapping), the visualization method (e.g. re-expression, sequencing), the scientific model (e.g. triangulation, classification) or the original data set (e.g. filtering). If temporal and cartographic animations are mixed, a temporal legend is needed to differentiate between world and presentation time [21].

Interactivity is becoming a main thread in exploring static and temporal visualizations (animations). It is ascertained that animations are more effective in information transmission if they can be temporally controlled by the viewer [16]. More interactive visualization methods (e.g. information hiding, linked brushing, data probing) do not increase effectivity but allow experienced users to enlarge their knowledge about the visualized data by exploring different features (information aspects). More complex and partial relationships or causal dependencies which are not so easily visible from analytical comparisons may be revealed. This forces the current development, evaluation and integration of many user-driven exploratory techniques into the presentation systems of animations and visualizations [6, 10, 1, 31].

3 Transmiting Interactive Animations on the Web

In the internet, the visualizations of the queried data set of Web-enabled databases as well as exploratory methods are required to be present at the client side. Therefore, some systems transmit the underlying spatio-temporal data set from the server to the client where specialized visualization and exploration software is used for presentation (*client-side data visualization*), like in [6] or MapObjects [13]. Although, this scenario allows many exploratory tools such as data manipulation or visualization parametrization, some drawbacks exist: on the client side, operational and scientific knowledge in visualizing data is needed if not the visualization is pre-defined or restricted; specialized software produces system-dependencies if not Java is used; no standard exists to transmit spatio-temporal data and entities on the internet (however, HDF exists for scientific data) causing software-dependencies; explorable visualizations can not be stored; and data distributors loose the control on the data set and its type of visualization.

A newly proposed approach is to attach the algorithm of a single visualization method as executable to the data set and send both to the user where this method is applied to the data [18]. But this requires special authoring systems, is limited to only one visualization method and depends on the operating system.

A better solution may be a graphical description of the visualized data which is generated on the server and sent to the client interface where simple interactive visualization tools may be hosted for presentation and exploration (*server-side data visualization*), like in [38, 2]. If this description is enhanced with information on time-references, change and database entities it allows more exploratory tools than in standard 3D-graphics (we assume spatio-temporal data as 3D+t-data).

Different graphical descriptions for interactive (explorable) animations and videos exist:

MPEG-4 is a recently standardized but not yet implemented video standard [28]. It supports the description of spatio-temporal objects via animated textures on animated 2D/3D-shapes in a synchronized spatio-temporal context and allows limited spatial and temporal navigation as well as further object interaction (e.g. hiding, moving). MPEG-4 is designed for storage and public presentation and needs the spatio-temporal graphics being encoded and then decoded for presentation which limits server-side data visualization on-demand.

Recently, new graphic file formats for (streamed) 3D-animations (e.g. Fluid3d [30], B3D [5], ScriptV [40]) were developed which are used only for presentation puposes with no or few features for user interaction. Also, new 3D-graphic libraries or programming languages (e.g. Java3D, Hypercosm with OMAR [17]) may be interesting for interactive WWW-based interfaces. But every automatically generated graphic need to be compiled on the server before transmitted to the client for execution because each graphic description is a program.

VRML is a standard graphic description format for object-oriented, interactive and animated 3D-graphics [42]. Visualizations of spatio-temporal data can be described as animated 3D-graphics but which until now can neither be interactively stopped nor run forwards nor backwards (possibly with different speeds) in the browser because the unique time concept inherent in VRML is a model of the "real time". This is also recognized in [29] where objects in VRML are extended with an additional time sensor to animate its time-variant behaviour on external events. But VRML offers the possibility to integrate new language concepts and to enhance the presentation systems with new interactive visualization methods, e.g. temporal navigation.

4 Analyzing Spatio-Temporal Data

Querying and interacting with databases means visualizing the data sets which are in the database and analyzing them by placing the data in different spatial, temporal or thematical context. This process stimulating mental cognition and understanding leads through several distinct levels which also characterizes the distinct functions of visualizations: ranging from exploration (discover unknown) and analysis (check, test and evaluate hypothesis) with more interactivity until synthesis (identify and combine different results) and presentation (convincingly communicate results to the public) [26]. On each level, the viewer follows different tasks to reach distinct interpretations of the data [19]: identify values, find values, compare values of one or multiple variables, recognize clusters and trends in one variable or correlation between multiple variables.

So, normally each specific user question requires a different visualization being the most efficient in transmiting the information or the features of the data to the viewer, i.e. being easily recognized and correctly interpreted [41]. This depends not only on the graphical design [4] but also on the user's knowledge, the visualization task and on the type of information being transferred [6]. In visual database interfaces providing access to the public, often, only basic background knowledge and a diffuse understanding of the data and visualization aims can be assumed. In this scenario, interactive visualization methods may help different features of the data to be made salient in *one* dynamic visualization which results in fewer server-client interactivity if server-side data visualization is used.

We first clarify which interpretations (information aspects) of the visualized spatio-temporal data may be revealed. This can be derived from the visualization tasks mentioned above. Then (in the next section), we examine possible exploratory tools which support the revelation of these data aspects.

The analysis of the spatial data component induces questions according to the spatial existence (if?), spatial location (where?) and spatial extend or distribution, i.e. shape (form?), size (how big?) and texture (variation) of objects and natural conditions. In relation to multiple objects spatial associations/dependencies (relationships?), spatial distance (how far?), spatial clustering (contiguity?) or spatial patterns (arrangements?) are of interest.

The analysis of the temporal data component may answer questions related to the existence and non-existence (if?), temporal location (when?), and duration (how long?) of single objects, facts and events. In relation to multiple objects temporal associations/dependencies (relationships?), temporal distance (how long?), temporal clustering (contiguity?), or temporal sequences/patterns (order?) may be investigated [21].

If both components are analyzed in combination other interpretations (information aspects) may be revealed [23, 7]. Commonlly, the user are interested in states and conditions anchored in the spatio-temporal context (what?, where? and when?) and in spatio-temporal change of objects or facts, especially in its existence (if?), location (when? and where?), type (how? about location: moving, rotating; about extend: growing, shrinking, deforming; about non-spatial: increasing, decreasing), pattern (how often?), rate (how fast?), sequence (which order?), trend, and in case of sets of changing objects in the associations/dependencies (about locations: approach, leave; about extend: merge, split, absorb, release), correlation or causal relationships.

5 Exploring Data with Interactive Visualizations

Interactive (exploratory) visualization techniques are implemented on different levels in visualization systems ranging from direct, easy to use techniques like changing 2D/3D-viewing parameters until indirect, algorithmic tools such as changing the parameters of data filters. The given information aspects may be revealed mostly by applying the exploratory tools distinguished in following categories [41]: viewing, encoding/filtering, reordering and algorithmic transformations. This categorization is motivated by the level of user knowledge and the data context needed to perform the tools.

For spatio-temporal data, first, interactive visualization methods which change the *viewing* of the visualization are required. This incorporates techniques for spatial navigation (zoom (with level of detail), pan, rotate), temporal navigation (start, direction, speed and resolution of animation) and presentation of a spatial and temporal overview with current position and current time, respectively. These techniques are wellknown, easy to use and exist in many presentation tools. Only the spatial 2D/3D and temporal context of the objects in the visualization are needed for execution.

For analyzing the existence of data in time and space as well as clusters and patterns, interactive tools for *encoding* and *filtering* different features of data are needed, e.g. linked brushing and focusing. They are performed on graphical objects or time-referenced states by selecting specific spatial, temporal or thematical values or value ranges. The operations are highlighting, colouring, hiding, filtering or aggregating the data. Additionally, tools which switch the visibility of graphical objects (visibility layers) or other data (e.g. transient labels, annotations, graphics, details) on some action (e.g. moving the mouse on an object) are useful. These tools need the structural information on objects and attributes.

In order to better compare multiple variables simultaneously in the spatial and temporal dimension, interactive tools for *reordering* the objects of the visualization are useful, such as spatial and temporal transformations (scale, move, rotate) or a change of the graphical design (colors, lights).

Other exploratory tools (*algorithmic transformations*) may also be of value, such as investigating the original data (e.g. data probing) or computing new data (e.g. height, distance, areas) through the graphics (*measuring*), changing the parameter of visualization methods (e.g. color-mapping) or the methods itself, e.g. re-expressions of time (linear vs. cyclic perspective) or space, or querying, filtering, transforming and manipulating the data from databases. But these tools are not considered for implementation because they need the underlying spatio-temporal data set to be performed and further knowledge in data visualization on the viewer side.

For better analysis of data sets of spatio-temporal databases exploratory tools of the first and second category are very important. However, in order to implement these tools in the client interface temporal and structural information of entity properties in the visualization description are needed.

6 Extension of VRML

In order to support the description of spatio-temporal entities and to implement many important exploratory tools for the analysis of spatio- temporal data, two new concepts are integrated in VRML: a) *valid time*, to permit the description of temporal behaviour and a temporal navigation functionality; and b) the *entity concept*, to leverage the abstraction level from single graphical object to conceptual entities known by the user from conceptual data modeling and querying databases. Both concepts are realized by creating new and redefining existing VRML-nodes via the PROTO-mechanism and implementing its behaviour in Java. After a short introduction to VRML, the concepts and the newly designed VRML-nodes are described. A subset is already proposed in VRML History [43] and deeply explained in [24]. All VRML-nodes are defined by their type and the list of attributes, each specified by the event type, data type, attribute name and default value.

6.1 Introduction to VRML

In VRML every visualization (3D-scene/world) is described by a set of objects (called nodes) which are instantiations of some pre-defined object types (e.g. geometries, materials, lights, sensors) and characterized by its attributes (called fields) and the graphical and logical behaviour. The objects are organized in a hierarchical scene graph via transformations (rotation, scale, translation) which determines the location and visibility of the objects. A special object type is the Viewpoint which specifies the position and viewing direction of the viewer in the world.

Utilizing an event mechanism, all object are able to send messages to other objects to modify their attributes. Sensor objects produce events on special system states (e.g. user interaction, clock) and are, therefore, the origin of action and animation in the world. An integrated interface to a programming language as well as a prototyping mechanism allow the definition of new object types with special behaviour.

6.2 Integrating Valid Time

The time concept already existing in VRML simulates approximately "real time" by generating unchangeably events as time passes. For a better distinction, it is called *system time* in this context.

In order to model temporal data in VRML-worlds, *valid time* is integrated as a new time dimension in VRML which specifies the time when a fact in the world was, is or will be valid. With valid time, each object in a VRML-world is transformed in a temporal object $o : (a_0, ..., a_n, vp)$ with the attributes a_i and the *valid period* vp. The valid period limits the *time of existence* or *validity* of the object to a specified time period $[t_0, t_n)$ where changes to object's characteristics may appear. Outside this period the object does not exist. By default, all objects are always valid, i.e. the valid period specifies $(-\infty, \infty)$.

Transaction time which is used in databases to model the time when a fact is recorded [35, 36] is not integrated in this proposal because this time concept is not relevant in data visualization, i.e. it has no special meaning in the visual understanding of processes and spatio-temporal data. However, visualization methods which show the difference between two versions of the same temporal data are possible.

The valid period as well as the times of change are defined by a list of consecutive times $(t_0, ..., t_n)$ via the *period* field of the newly designed *ValidPeriod* node. The n times $t_0, t_1, ..., t_n$ partition the time domain $(-\infty, +\infty)$ into $n+2$ subintervals called *version time periods* where each of them may be associated with an object or attribute *version*, except the first and last one.

```
ValidPeriod {
  exposedField  SFBool  isCyclic        FALSE
  exposedField  MFTime  period          []
  exposedField  SFTime  periodOffset    0.0
  exposedField  SFNode  reference       NULL
  exposedField  SFBool  fromStartOfRef  TRUE
}
```

All time values are specified as floating point numbers relative to a node-specific, local time origin which may be the global time origin or the starting/ending time of another already specified valid period (via field *reference*) (for temporal causal dependencies), moved by an offset (field *periodOffset*). This model gives much flexibility in defining absolute, relative and cyclic times where manipulations, e.g. moving an object in time, can easily be realized. Additionally, it avoids parsing different date formats and calendars and allows some kind of independence of the time unit (e.g years, days, seconds) and of the resolution of discrete time values.

6.3 Temporal Data Modeling Concepts

The research in spatio-temporal data modeling has identified different types of spatio-temporal data [32, 39, 12]: spatial objects a) without change (*time-invariant objects*), b) which only move, c) which only change its extent or d) both (*time-variant objects*) and e) changing spatial fields which despite the differences in database modelling may also be represented as time-variant objects in visualizations [39].

Time-invariant objects possess only one non-changeable representation valid during the given valid period. Any time-variant object possesses different time-referenced representations (*versions*) during its valid period which compose the *version history* of this object. The description of these representation is done with the known concepts of temporal data modeling [12, 35, 36], i.e. time-(in)variant attributes, discrete(event-oriented)/step-wise/linear versioning (how do attributes evolve over time?), and tuple-/attribute-timestamping. These results in the following newly designed nodes which may be used in connection with the existing graphical VRML-nodes (i.e. shape, appearance, lights, environment, sensors).

Limited Validity and Time-Referenced Object Versions
The new grouping node *History* is designed to limit the validity of its versions. Each version v_i specifies a local temporal VRML-world whose validity is given by the associated version time period $[t_i, t_{i+1})$ specified in the field *period* of the ValidPeriod node which is the value of the field *validPeriod* of this node type.

```
History {
  exposedField  SFNode  validPeriod  NULL
  exposedField  MFNode  version      []
}
```

This node may be used: a) to limit the validity of a time-invariant object defined as the first version; or b) to specify a time-variant object whose representation is changing in a step-wise mode, i.e. consists of a series of constant, time-referenced object versions.

Time-Variant Attributes

Attributes of time-variant objects may be time-invariant or time-variant. Time-variant nominal attributes (of textual, boolean, image or node type) have to be modelled via time-referenced object versions. Numerical time-variant attributes as well as attributes for coordinates and colors, no matter whether changing instantly or continuously, are modelled via time series decribed in the next subsection.

Attributes of VRML's graphical shapes for the geometry and texture object are handled separately because different types of these objects exist which may change in valid time (geometry: e.g. Box, Cone, IndexedFaceSet; texture: e.g. MovieTexture, PixelTexture). The *HistoryGeometry* and *HistoryTexture* nodes are designed to handle discrete or step-wise versioned geometry and texture types respectively.

```
HistoryGeometry/Texture {
  exposedField  SFNode  validPeriod       NULL
  exposedField  MFNode  geometry/texture  []
}
```

Both nodes allow a possible later implementation of interpolation (i.e. morphing) algorithms for geometry and texture types.

Time Series of Numerical Attributes

Time series conceptually describe a list of pairs of attribute and associated time values which are the base points of a somehow interpolated attribute function. For flexibility, this is divided into the definition of two nodes: the *time value* (ValidTimeSensor) and the *attribute value* definition (Interpolator). The times of the base points list are specified in the field *validPeriod* of the *ValidTimeSensor* node, the attribute values in the different data type-dependent VRML-Interpolator (e.g. Normal, Coordinate, Scalar, Color) which uses linear interpolation to compute new values.

```
ValidTimeSensor {
  exposedField  SFBool   enabled           TRUE
  exposedField  SFNode   validPeriod       NULL
  eventOut      SFFloat  fraction_changed
  eventOut      SFBool   isActive
  eventOut      SFTime   validTime
}
```

Both nodes and the specific numerical attribute are linked by employing VRML's event/message routing technique. On every temporal navigation action, i.e. the viewer moves temporally in the given time periods of the time value definition, the ValidTimeSensor produces events which are sent to the Interpolator and then to the attribute to change its value. Quadratic, cubic or other functional descriptions may reduce the amount of base points needed to approximate a given time-variant behaviour in the interpolator, but linear interpolation is much easier to model, faster in execution and even used in databases [12].

6.4 Temporal Navigation

In order to investigate the temporal dimension of the visualization a specific navigation method is needed. Like in 3D-navigation, a "viewpoint" is integrated in the valid time dimension which is responsible for the current presentation of the temporal world. Only those objects and their representation are shown which are valid at the time given by the current viewpoint (like a time-slice or snapshot of the temporal world). The extended *Viewpoint* node specifies spatial and temporal key locations given by the fields *position*, *orientation*, and *valid time*.

```
Viewpoint {
  ...
  exposedField SFRotation orientation 0 0 1 0
  exposedField SFVec3f    position    0 0 10
  exposedField SFTime     validTime   0.0       # added
  ...
}
```

Changing the valid time allows the viewer to move through the temporal dimension of the world (i.e. doing a time journey), interactively via the new navigation paradigms or automatically by animating the field *validTime* of the Viewpoint node via the TimeInterpolator. The *TimeInterpolator* node interpolates linearly among a list of time values specified in the field *keyValue*.

```
TimeInterpolator {
  eventIn       SFFloat  set_fraction
  exposedField  MFFloat  key           []
  exposedField  MFTime   keyValue      []
  eventOut      SFTime   value_changed
}
```

The navigation paradigm describes the way the position, orientation *and* valid time of the current viewpoint may be changed. Two new navigation paradigms are introduced which can be set in the field *type* of the *NavigationInfo* node to explore the temporal domain of the visualization: TIME_WALK and TIME_STUDY. TIME_WALK allows walking with changing speed forward or backward in the temporal domain, TIME_STUDY allows to investigate the local surrounding of the current valid time.

The temporal navigation is performed in parallel to the spatial navigation and globally to the scene, i.e. synchronously to all objects defined in the temporal VRML-world. This is more intuitive for the viewer and the presented scene is easier to be understood and recognized than if the local valid time of each object is changed independently like in [14]. Instead, if some applications (e.g. scheduling, simulations) need to view simultaneously objects at different valid times, it is easier to temporally move the valid period of these objects along the valid time dimension.

6.5 Temporal Legend and Temporal Data Presentation

A temporal legend is developed to facilitate two main functions [21]: a) interacting with the temporal domain of the visualization and b) interpreting the shown graphics in the temporal sense, i.e. distinguishing between a cartographic and temporal animation. In order to perform these functions the legend consists of two components (Fig. 1):

- the *controls* allow to navigate interactively in the temporal domain according to the navigation paradigms, to jump to a newly set current time value, and to start an animation with user-defined speed and direction over a specified period;
- the *time display* shows textually the current valid time used for temporal presentation as, e.g. a date, numbers, or simulation ticks since a pre-defined time origin.

For temporal analysis, a presentation of the temporal data components may be valuable. This is done in a *temporal overview* which renders the times of existence (valid period) of the temporal graphical objects in relation to the current valid times over the current visible part of the time line. This is important for event-oriented graphical objects with a moment-like valid period which, generally, are hard to be discovered while navigating temporally because they often fall between two presented snapshots. The rendered valid periods are enhanced with symbolized information, such as the base points of change, the type of versions, and the information if the time definition is cyclic or linked to another valid period. Additionally, a standard or user-defined cyclic reference devision of the time line (e.g. in years, weeks) is displayed for a better orientation in time.

In this implementation the legend and temporal overview are integrated in the browser's display to facilitate a link to the currently presented VRML-world.

Two further exploratory methods are implemented: temporal focusing and temporal brushing. In *temporal focusing* all events and objects are shown which are valid at a user-defineable period. In *temporal brushing* several periods of the time line may be selected which then constitute the new temporal basis for temporal navigation. This technique filters certain periods in time and is often used with a cyclic perspective of time to analyse long-time trends of cyclic phenomena (e.g. wheather cycles) [16].

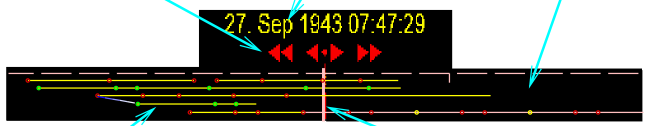

Fig. 1. Appearance of the temporal legend and temporal overview

6.6 Exploring Spatial-Temporal Entities

The user's view on data stored in databases is heavily influenced by conceptual data modeling whose main constructs are entities with its attributes and relationships. Even spatial data are modeled via the entity concept [39]. The new node *Entity* (such as proposed in [8]) is designed to group a set of VRML-nodes forming a spatio-temporal database entity. All entities created by this node type are grouped in layers and listed in a "entity"-window which can additionally be displayed in the user interface.

```
Entity {
   exposedField   SFString    id             ""
   exposedField   SFString    layer          ""
   evenOut        SFString    selectedId
   eventIn        SFBool      isSelected
   exposedField   SFBool      isVisible      TRUE
   exposedField   SFBool      isHighlighted  FALSE
   exposedField   MFNode      shape          []
   exposedField   MFNode      attribute      []
}
```

Program logic, located internally or externally of a VRML-world, can be used to switch visibility (field *isVisible*), to highlight (field *isHighlighted*) or to select (field *isSelected*) one entity. In the latter case it causes the entity to send its id (via the field *selectedId*) back to the scene where this information may be used to interact with a database [8]. The field *shape* contains the possibly time-variant shape, the field *attribute* the possibly time-variant detail informations of the entity which are displayed textually in an additional "entity details"-window and which are also sensitive on temporal navigation. Detailed information may be meta data or non-spatial entity characteristics. The following node types are allowed in that field which have no graphical effect.

```
NumericalAttribute {
   exposedField SFString   name
   exposedField SFNode     validPeriod
```

```
   exposedField MFInt32    versionType
   exposedField MFFloat    value
   eventOut     SFFloat    value_changed
}
```

Similar node types exist for temporal textual (*TextAttribute*) and temporal time attributes (*TimeAttribute*). The field *value_changed* returns the currently valid value of the time-variant attribute back to the scene while navigating temporally.

Based on this entity concept, some of the exploratory tools mentioned in the preceding section are implemented to allow private investigation of spatio-temporal data. Layers of entities can be made interactively (un)visible to reduce the amount of displayed information and facilitate the concentration on certain features (*information hiding*). A *linked brushing* method is implemented between the "entity"-window, the presented scene and the temporal overview to locate and distinquish entities in different context. In *temporal comparison* the viewer apply a temporal transformation (move, scale) to the time definition of one single entity to simultaneously compare and find correlations in the spatio-temporal behaviour of entities which are normally temporally distanced.

7 Applications

Two example applications show how the proposed VRML-nodes are used to explore spatio-temporal data sets for decision support: visualizing temporal finite-element method (FEM) data in water management and temporal GIS-data in urban planning. Each visualization is published in the internet [43], stored as a normal VRML-file, and can be presented and explored with the described tools using a standard VRML-Browser and the implementation of the proposed node types [43].

7.1 Visualizing Temporal FEM-Data

In water management, FEM-simulation are often used to solve optimization problems in order to support the decision process [15]: In a high water region the ground water head has to be kept below a certain level utilizing several wells whose yearly pumping cost has to be minimized according to the temporal flow and rise of new ground water. In Fig. 2 (visualized with Netscape Navigator and Cosmo's CosmoPlayer), several snapshots of two ground water heads, i.e. the original, already practiced (black) and the optimized (dark grey) pumping rate, over 4 years are shown including the location of the well, some control points (dark grey bars), different depth levels (almost transparent grey rectangle) and the upper level (dark grey rectangle). Aspects of anlaysis are the spatio-temporal behaviour of the ground water head at times when the pumping rate changes, the well's spatial influence on the ground water flow process, and correlations and differences between the two ground water heads.

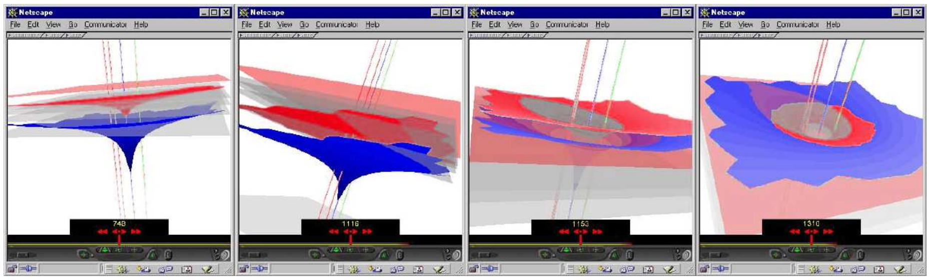

Fig. 2. Snapshots showing two temporal ground water heads

The VRML-file contains two resulting FEM-simulation data sets (106 FEs) mapped onto colored triangle irregular networks (TIN) at 160 different valid times. The time-referenced TINs are modelled as the base points of a geometrical time series which during presentation yields a continuously changing shape by linear interpolation. The VRML-file's size is about 10 MB, compressed 1.5 MB.

7.2 Visualizing Temporal GIS-Data

In this example the proposed VRML-nodes are used to visualize a set of synthetically produced temporal GIS-data representing the possible data of an urban planning application. i.e. constructions and natural objects (e.g. buildings, places, streets, parcs, trees) and events (e.g. reconstruction, demolition) of the historical development of a city (Fig. 3). Aspects of analysis are the visualization of historical maps and city views, the architectural and social development of certain city districts as well as periods of much construction activity.

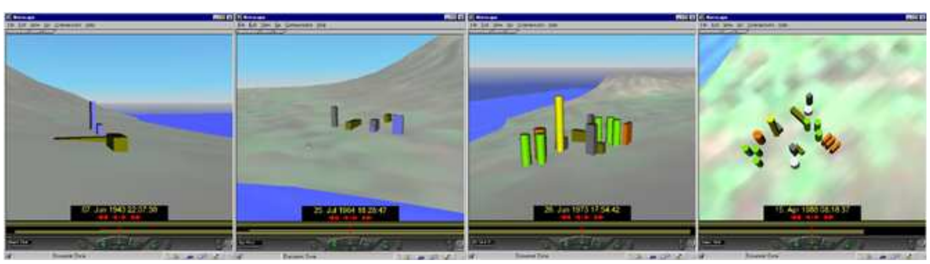

Fig. 3. Snapshots showing historical views of a city

The data stored as the temporal visualization are sets of entities with time-variant geometries and appearances which exist at different periods in time. The time-variant sea level changes according to the tides. Note, that the waves are animated not by valid but by system time which can not be changed by the viewer. This time concept can be used for further cartographic techniques, such as object flashing for getting attention, time-variant symbols (e.g. rain symbol) or change in object's representation (e.g. re-expression). The VRML-file's size is about 38 kB.

8 Discussion and Future Work

One big advantage of our approach is that each interactive, temporal visualization can be stored in a file and integrated in a Web-page for later presentation or publication in the WWW, e.g. for knowledge transmission, cooperative work, marketing or documentation. The exploratory tools are kept with the new language constructs.

The document description language XML may be the new emerging standard for documents containing structured informations in the WWW. Because graphics are always a part of documents, their file formats should be integrated into this language to allow a tight interaction. X3D is the proposal of the VRML-community to reach this aim [44]. One future plan is the integration of the node types proposed in this paper in this process to feature interactive, temporal visualizations.

Many more exploratory tools are mentioned but not yet implemented, e.g. spatial focusing, spatial transformation, focusing and brushing based on the detail information, transient labels over visualized data with informations, such as names, detail informations, time series diagrams or non-spatial time-variant attributes.

In this implementation, temporal focusing involves a simple temporal aggregation technique which is adequate for events. For temporal graphical objects with longer valid periods only the first version is taken until now which is not adequate. Aggregation techniques which include the time-dependency of the spatial and non-spatial data are more appropriate.

Chaotic or non-linear behaviour of objects or object sets, especially, if the components start and cease to exist at different times (e.g. particles, smoke, explosion), is badly described by linear or step-wise functions. More special, i.e. functional, stochastic or evolutionary descriptions would better match these needs.

Contrary to this proposal, the user handles time information in form of readable dates and times based on specific calendars. For efficiency, it is assumed that authoring and presentation systems perform these transformations.

One inevitable disadvantage of spatio-temporal data is the size of the description. Generally, the size of each temporal object or attribute description is determined by the non-temporal description multiplied with the number of time-referenced changes. In relation to static worlds this increases the size of the VRML-file and download time multiple times. There are various approaches to reduce the file size: Authors can reduce the spatial and temporal description of less important objects by using linear interpolation or aggregation methods, i.e. reducing the number of polygons/vertices and times forming the base points of the temporal description. Content-independent approaches discussed by the

VRML-community are: object referencing, file compression, binary file formats or content streaming. Especially content streaming seems to be very useful for temporal data: the first snapshot can already be presented while the others are still be loaded.

The size of the spatio-temporal description leads also to a huge scene graph and object storage in the browser. If no temporal navigation is performed this has no effect on the processing time for each image (i.e. frame rate). However, on each temporal navigation action, performed interactively or via animation, the list of Validperiod nodes is parsed for validity at that specific time. If valid, the specific version of the object is determined which then changes the spatial appearance of the object in the scene graph. These steps increase processing time. So, reusing ValidPeriod and ValidTimeSensor nodes for multiple temporal attribute description and reducing the number of versions of each object is good practise in describing visualizations. Further techniques may be a level of detail for the spatial dimension of spatio-temporal objects and a level of interest for the temporal dimension to reduce the number of temporal objects parsed.

A browser-specific enhancement concerns the scene graph. Many objects do not need to be processed because they are not visible in the current view depending on the current viewpoint position, orientation and valid time. For static temporal scenes, a spatio-temporal access method may be very usable to reach the relevant set of objects in a short time [37].

9 Conclusion

This paper outlines our approach for a graphic description language for a new type of visualization, i.e. interactive, temporal visualizations (animations). Since 3D-graphics are not adequate for displaying data sets of spatio-temporal databases both in a spatial and temporal context, we propose to extend VRML with a new time dimension (valid time) for the description of temporal object informations, and an entity concept to organize graphical data in structures known from database modeling. With these constructs the WWW-based presentation systems or browsers (like WWW-based visual database interfaces) can offer a rich set of simple and easy to use exploratory tools during presentation to the public end user which can be applied for an improved, individual analysis of the distinct features (information aspects) of spatio-temporal data.

References

1. Andrienko, G.L., Andrienko, N.V.: Interactive Maps for Visual Data Exploration. On: ICA Commission on Visualization Meeting, Warshaw (1998)
2. Advanced Visual Systems Inc.: AVS/Express. At: http://www.avs.com
3. Bernard, L., Schmidt, B., Streit, U., Uhlenküken, C.: Managing, Modeling and Visualizing High-Dimensional Spatio-temporal Data in an Integrated System. In: GeoInformatica 2(1). Kluwer Academics Publishers (1998) 59-77
4. Bertin, J.: Semiology of Graphics: Diagrams, Networks, Maps. University of Wisconsin Press (1983)

5. Brilliant Digital Entertainment Inc.: B3D. At: http://www.bde3d.com (1999)
6. Buziek, G., Hatger, C.: Interactive animation and digital cartometry by VRML 2.0 and Java within a temporal environmental model on the basis of a DTM of the Elbe estuary and a 12 hour tide period. On: ICA Commission on Visualization Meeting, Warshaw (1998)
7. Claramunt, C., Thériault, M.: Managing Time in GIS: an Event-Oriented Approach. In: Clifford, J., Tuzhilin, A. (eds.): Recent Advances in Temporal Databases. Springer-Verlag (1995) 23-42
8. Coors, V., Jung, V.: Using VRML as an Interface to the 3D Data Warehouse. In: Proc. of 3. Symp. on VRML. ACM Press (1998)
9. Dransch, D.: Computer-Animationen in der Kartographie - Theorie und Praxis. Springer-Verlag (1997)
10. Dykes, J.A.: Exploring Spatial data Representation with Dynamic Graphics. In: Computer & Geosciences 23(4). Elsevier Science (1997) 345-370
11. Edsall, R., Peuquet, D.: A Graphical User Interface for the Integration of Time into GIS. Research Paper of The Apoala Project. Pennsylvania State University. Online Publication: http://www.geog.psu.edu/apoala/newPaper.html (1997)
12. Erwig, M., Güting, R.H., Schneider, M., Vazirgiannis, M.: Spatio-temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases. In: Proc. 1. CHOROCHRONOS Workshop on Spatio-Temporal Database Systems. ChoroChronos TR CH-97-02 (1997) 79-86
13. Esri Inc.: MapObjects. At: http://www.esri.com
14. GRAPE Manual. Inst. of Applied Mathematics. Univ. of Bonn and Univ. of Freiburg (1997)
15. Grauer, M., Barth, T., Kaden, S., Michels, I.: Decision Support and distributed Computing in Groundwater Management. In: Savic, D., Walters, G. (eds.): Water Industry Systems: Modelling and Optimization Applications. Research Studies Press Ltd. (1999)
16. Harrower, M., MacEachren, A.M.: Exploratory Data Analysis and map animation: Using temporal brushing and focusing to facilitate learning about global wheather. On: ICA Commission on Visualization Meeting, Warshaw (1998)
17. Hypercosm, Inc.: The Hypercosm 3D Graphics System. Rendering, Animation, Simulation, and VR System. At: http://www.hypercosm.com/ (1999)
18. Jern, M.: Information Visualization on the Web. On: CODATA Euro-American Workshop on Visualization of Information and Data (1997)
19. Jung, V.: Fuzzy effectiveness evaluation for intelligent user interfaces to GIS visualization. In: Proc. of 4th ACM Int. Workshop on Advances in GIS. ACM Press (1996) 157-164
20. Johnson, I.: Mapping the fourth dimension: the TimeMap project. In: Proc. of the 25th Conf. Computer Applications in Archaelogy (1997)
21. Kraak, M.-J., Edsall, R., MacEachren, A.M.: Cartographic Animation and Legends for Temporal Maps: Exploration and/or Interaction. In: Proc. of 18th Int. Cartographic Conf. in Stockholm (1997) 253-261
22. Kraak, M.-J., MacEachren, A.M.: Visualization of the Temporal Component of spatial Data. In: Waugh, T.C., Healey, R.G. (eds.): Proc. of 6. Int. Symp. on Spatial Data Handling. Taylor & Francis (1994) 391-409
23. Langran, G.: Analyzing and Generalizing Temporal Geographic Information. In: Proc. of GIS/LIS 1. ACSM-ASPRS-UIRSA-AM/FM (1993) 379-388
24. Luttermann, H., Grauer, M.: VRML History: Storing and Browsing Temporal 3D-Worlds. In: Proc. of 4. Symp. on VRML. ACM Press (1999) 153-160

25. MacEachren, A.M.: Time as a Cartographic Variable. In: Hearnshaw, K.M., Unwin, D.J. (eds.): Visualization in GIS. John Wiley & Sons (1994) 115-130
26. MacEachren, A.M., M.-J. Kraak, M.-J.: Exploratory Cartographic Visualization: Advancing the Agenda. In: Computers & Geosciences 23(4). Elsevier Science (1997) 335-343
27. MacEachren, A.M., Polsky, C., Haug, D., Brown, D., Boscoe, F., Beedasy, J., Pickle, L., Marrara, M.: Visualizing spatial Relationships among Health, Environmental, and Demographic Statistics: Interface Design Issues. In: Proc. of 18th Int. Cartographic Conf. in Stockholm (1997) 880-887
28. Moving Picture Experts Group. At: http://www.mpeg.org
29. Mirbel, I., Pernici, B., Vazirgiannis, M.: Modeling VRML Worlds with ECA Rules. Slides from the Dagstuhl Seminar: Integrating Spatial and Temporal Databases (1998)
30. Oz.com: Fluid3D. At: http://www.oz.com/fluid3d (1998)
31. Peterson, M.P.: Interactive and Animated Cartography. Prentice Hall (1995)
32. Pfoser, D., Tryfona, N.: Requirements, Definitions and Notations for Spatio-temporal Application Environments. ChoroChronos TR CH-98-09 (1998)
33. Shepherd, I.D.H.: Putting Time on the Map: Dynamic Display in Data Visualization and GIS. In: Fischer, P. (ed.): Innovation in GIS 2. Taylor & Francis (1995)
34. Schroeder, W.J., Martin, K.M., Lorensen, W.E.: The Visualization Toolkit: An object-Oriented Approach To 3D Graphics. Prentice Hall (1997)
35. Snodgras, R.T. (Ed.): The TSQL2 Temporal Query Language. Kluwer Academic Press (1995)
36. Tansel, A.U., Clifford, J., Gadia, S.K., Jajodia, S., Segev, A., Snodgras, R.T. (eds.): Temporal Databases: Theory, Design, and Implementation. Benjamin/Cummings (1993)
37. Theodoridis, Y.: Indexing Methods for Spatiotemporal Databases. In: Proc. 1. CHOROCHRONOS Workshop on Spatio-Temporal Database Systems. ChoroChronos TR CH-97-02 (1997) 37-42
38. Theodoulidis, B., Vakaloudis, A.: A Data Model and a Query Language for 3D Spatiotemporal Information. In: Proc. 1. CHOROCHRONOS Workshop on Spatio-Temporal Database Systems. ChoroChronos TR CH-97-02 (1997) 17-25
39. Tryfona, N., Jensen, C.S.: Conceptual Data Modeling for Spatiotemporal Applications. ChoroChronos TR CH-98-08 (1998)
40. Visviva Software Inc.: ScriptV. At: http://www.visviva.com (1998)
41. Tweedie, L.: Characterizing Interactive Externalizations. In: Proc. of CHI'97. ACM Press (1997)
42. VRML Architecture Group (VAG): The Virtual Reality Modeling Language Specifications ISO/IEC DIS 14772-1. At: http://www.web3d.org/Specifications/VRML97/ (1997)
43. University of Siegen: The VRML History Homepage. At: http://www-winfo.uni-siegen.de/vrmlHistory/docs/index.html (1998)
44. Web3D Consortium: X3D. At: http://www.web3d.org (1999)