Continuum: Designing Timelines for Hierarchies, Relationships and Scale

Paul André, Max L. Wilson, Alistair Russell, Daniel A. Smith, Alistair Owens and m.c. schraefel
School of Electronics and Computer Science
University of Southampton, UK
{pa2, mlw05r, ar5, das05r, ao, mc}
@ecs.soton.ac.uk

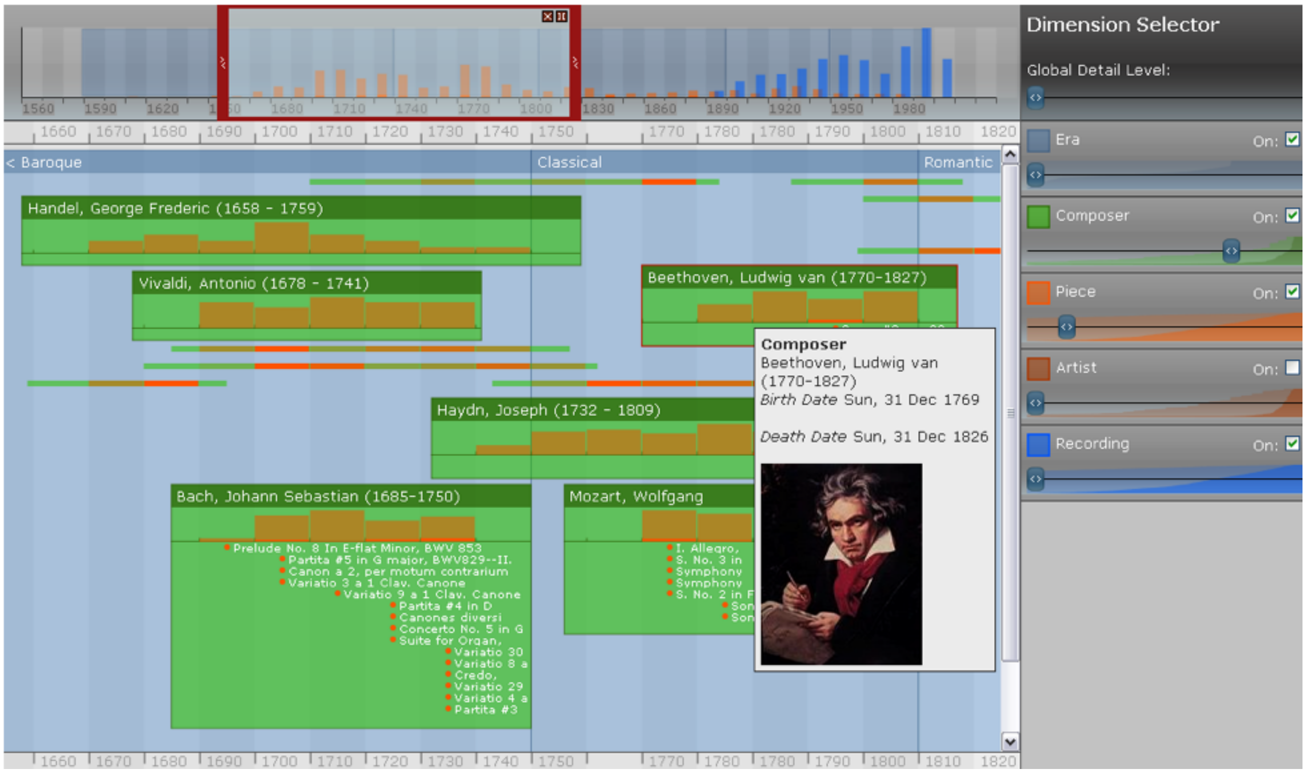

Figure 1: Continuum, a timeline visualisation tool for representing faceted temporal data.

ABSTRACT

Temporal events, while often discrete, also have interesting relationships within and across times: larger events are often collections of smaller more discrete events (battles within wars; artists' works within a form); events at one point also have correlations with events at other points (a play written in one period is related to its performance over a period of time). Most temporal visualisations, however, only represent discrete data points or single data types along a single timeline: this event started here and ended there; this work was published at this time; this tag was popular for this period. In order to represent richer, faceted attributes of temporal events, we present Continuum. Continuum enables hierarchical relationships in temporal data to be represented and explored; it enables relationships between events across periods to be expressed, and in particular it enables user-determined control over the level of detail of any facet of interest so that the person using the system can determine a focus point, no matter the level of zoom over the temporal space. We present the factors motivating our approach, our evaluation and implementation of this new visualisation which makes it easy for anyone to apply this interface to rich, large-scale datasets with temporal data.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design, Human Factors

Keywords: Timeline, information visualisation, user interfaces, hierarchical relationships

UIST'07, October 7–10, 2007, Newport, Rhode Island, USA.

INTRODUCTION

Rich information environments, like mSpace [24] and other faceted browsers [9] make it possible to view high dimensional data from multiple perspectives. In a classical music space for instance, the dimensions in the domain may include Era, Composer, Instrument and Piece. In mSpace, these dimensions can be organized into a dynamic hierarchy from left to right across the screen. Placed in that order, a selection in Era filters the remaining dimensions.

These dimensions can be reorganised so that Instrument, for instance, can be moved to the top of the hierarchy, then Era, Composer, Piece. Selecting something in Instrument (such as Cello) filters what appears now in the associated Eras, Composers, and Pieces. Each of these dimensions has a temporal component to its instances: specific instruments were developed at particular periods; composers lived for certain times; pieces were composed or published at certain dates. Other potential dimensions within this space have interesting relationships which also have temporal dimensions, such as performances and recordings of works relations to compositions. Faceted temporal browsing therefore would make it possible to foreground: (1) hierarchical relationships in the data (such as pieces within composers within eras) (2) relationships among dimensions in the data (how recording artists in one period relate to compositions in another). A challenge of such an approach would be to ensure that (3) information in view remains meaningful even at scale (not letting dozens of pieces by one composer over time become an indistinguishable blob).

Current timeline visualisations are unable to support these rich temporal relationships foregrounded by faceted browsing. The Simile Timeline [25], a popular Web2.0 widget, is highly interactive but only supports viewing (potentially hierarchical or related) events or spans of time in an unrelated manner. A foundational timeline example, LifeLines [22], is able to display facets of a hierarchy, but only on individual rows within the visualisation. Relationships among items between rows are only available while clicking an individual item, which shows relationships in other facets. These visualisations are also unable to scale effectively to large datasets: in an effort to make all information visible, detailed information is absorbed such that the most that can be clearly derived from this overview is "something happened at this time."

To address these issues of visualising temporal information (a) within dynamic hierarchies, (b) across-concept relationships/associations, and (c) in large scale overviews with meaningful detail, we propose Continuum (Figure 1), a Web2.0 application for visualising faceted temporal data. In the following sections, we describe related work in temporal visualisation, the interaction design for our approach, an overview of the key points of the implementation, the evaluation of the approach, a discussion of our results, and our contributions and plans for future work.

RELATED WORK

Timeline visualisations are a popular approach to representing temporal data. Usually along a uniform axis, data points are plotted so that their relative time associations can be viewed, supporting the comparison of different parts of data. Tufte [26] points to the strengths of a timeline visualisation (of a New York weather summary for 1980) that "successfully organises a large collection of numbers, makes comparisons between different parts of the data, and tells a story". Although there are such examples of sophisticated visualisations as referenced by Tufte, they are generally manually-drawn, carefully-studied, and domain-specific timelines. Current automatic visualisation research has largely focused on flat relationships between temporal events, and fails to support the needs of developing rich information environments.

Early work by Cousins and Kahn [7] in 1991 presented a formal mathematical definition for representing complex temporal data on a timeline. The system allowed for five operations to control the content of the visualisation: slice, filter, overlay, new and add. Kumar et al. [15] noted that this system was too rigid and had limitations for incorporating certain types of information. Subsequently, they stated what was needed was a "model that expresses the process of creating and presenting knowledge-rich content in a flexible manner." They developed the Interactive Timeline Editing and Review (ITER) framework within which generalised timelines can be created and viewed. Though the purpose of our work is not to develop such a model, we are interested in their approach to visualising and controlling the representation. The developed prototype is one of few examples that promote the display of relationships, although this is in a fairly obfuscated way due to the requirement to zoom, and thus lose context, choosing which relationships to view through a menu, or having to define derived attributes on the fly. Chittaro and Combi [6] furthered research into the visualisation of temporal information and proposed three potential solutions that account for representing relations, precise end-points and possible on-going periods. However, the focus was on different visual vocabularies rather than the design of a complete visualisation tool.

The Perspective Wall [16] is a timeline visualisation tool that integrates detail and context views of a linear information space (such as time) by 'folding': a panel in the centre gives a detail view, and two perspective panels on either side relate context. The detail+context (or focus+context) concept is an important consideration in design [8]. This is one approach to zooming, where the viewer zooms in on information by moving the timeline so that the entity is on the centre panel. However, this is limited to focusing on specific periods of time, rather than allowing filtering by class or specific items of information, and so still has limitations on usability with large scale datasets.

Richter et al. [23] address the need for keeping focus+context by using a multi-scale timeline slider in which new timelines are spawned by focusing on a region of the previous timeline. This also allows the user to focus on periods of time, but unlike the perspective wall, it simultaneously displays the same information as part of a uniform overview. However, the different levels of zoom do not convey more or less temporal information, but more or less metadata. Subsequently, it still has limitations of scale. Another timeline tool, employed in Jog and Shneiderman's FilmFinder [12], demonstrates a zooming starfield display. Whilst the focus of the paper is not on the temporal aspects of the data, the X-axis denotes time, and by presenting an overview and allowing the user to zoom and filter areas of interest, it presents a similar concept. Interestingly, the data they deal with (films with attributes of title, director, genre, actors) could be seen as a hierarchy, yet this is not explored.

A slightly different approach to displaying time was taken by the TimeSlider [13], in which the ends of the time scale are exponential, allowing an extensive time range to be displayed in a small area. While this approach also allows for zooming and context, Richter et al. suggest that the non-linear representation of time has negative effects on the interpretation of data [23].

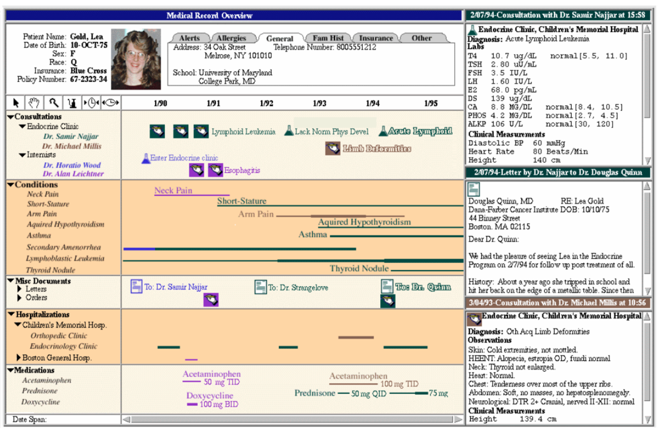

Figure 2: The LifeLines system. Related entities are shown, for example each doctor and their consultations, medications are shown by a colour, but the user must themselves link those actions by panning vertically.

The LifeLines [22, 21] system for visualising personal histories is the first to bring together the full gamut of problems facing timelines: the overview, hierarchy, rescaling, interrelationships, and layout issues. Though the paper addresses scale for the intended domain, we seek to address scale on a much larger scale, (displaying information about 1,000 or 10,000 records rather than 30). As noted earlier, LifeLines is able to display hierarchies and relationships, but with only colour coding to help, the user must make the links between related entities themselves, as shown in Figure 2. The concept of semantic zooming [4] is key to our work, allowing a meaningful overview at each scale.

Jensen [11], in SemTime, began to address relationships and hierarchy, and used time-independent stacking of multiple timelines to show relationships between events. Hierarchical timelines are also considered, although differently to our approach. SemTime allows the expansion of, for example, the Seven Years War item, into a sub-timeline. Events within that item are then made visible. However, SemTime does not discuss taxonomic hierarchies, which we focus on, and the ability to dynamically reorder hierarchies.

Brodbeck and Girardin [5] present a preliminary look at TrendDesign, a tool to represent and evaluate large amounts of time-dependent measured data. They use a bifocal lens [1] as a semantic zoom (interestingly, using histograms) to provide access to the appropriate representation at different timescales.

Bade et al. [3] extend the LifeLines qualitative/quantitative scales by introducing colour- and height-coded timeline representations. Integrating the concepts of pan+zoom, focus+context, and overview+detail, 3 stacked connected timelines are displayed, from a fixed overview, and through selecting sub-ranges and defining temporal bounds, filtering to more detail.

The most recent and widely available work in this area (it has been successfully open sourced) is the Simile Timeline [25]. Like Richter et al. a uniform overview timeline presents context while a more detailed view focuses on a specified area within the time space. Hierarchy and relationships are not dealt with explicitly, but permitted to certain extents through controls such as colour, but only at one level at a time. For example, classical composers and compositions could be colour coded with the eras, but then compositions cannot also be colour coded to composers at the same time. Correspondence with the lead developer indicates the Simile Timeline has not been engineered to deal with significant scale; visualising a dataset would be an issue above approximately 700 items.

Gantt charts, in various commercial solutions, have advanced timeline visualisation and manipulation capabilities that often extend (and inter-operate) with a package such as MS Project [19]. Artemis [2] has the ability to show inter-project relationships, to compare a selected project against a baseline, and has a combined bar-chart and histogram display. The histogram display, that Continuum also uses, is utilised in OmniPlan [20] to give quantitative detail (of resource allocation) in an overview. Dependencies, constraints, and filtering tasks by resources are also available; the idea of visibility filters is picked up in ILOG [10], along with sorting facilities and collapsing/expanding views. MinuteMan [18] provides an outline view, with the ability to zoom or display in detail one task.

Some of these innovations, such as the histogram view, are utilised in Continuum, and some, such as filters and sorting, represent areas we touch on but would like to explore in more detail in future work. However, we would argue that Continuum differs from Gantt in the following ways:

1) We visualise high-dimensional data by flattening it onto a projection which then becomes a temporary hierarchy with flexible associations: dimensions can be reorganised, added, subtracted. Most Gantts have only a fixed hierarchy.

2) Gantts allow zooming, but this normally displays more metadata and not a different representation. By using semantic zooming, we provide a meaningful representation at all levels, e.g. a histogram or piece list.

3) Gantts group rather than summarise. In expanding a group all, (for example), Admin tasks are detailed. This is useful for that domain, but we wanted something different, by summarising we provide a visualisation that immediately conveys quantitative information.

4) An ongoing thread on Tufte's site [27] shows many companies print their expansive Gantt charts, make alterations, and print weekly. Our design addresses the issues that render serious Gantt charts impractical: summarising, semantic zoom, and the overview panel giving context+detail.

APPLICATION DESIGN

In order to represent faceted temporal data, we needed to be able to visualize temporal information (a) within dynamic hierarchies, (b) across-concept relationships/associations, and (c) in large scale overviews with meaningful detail. We iterated on numerous paper and Flash prototypes via cognitive walkthroughs of designs with participants. Drawing from these investigations, the following key design attributes were identified: 1) at any level of zoom, something immediately useful must be conveyed; 2) where information is minimised to reduce clutter, either by the system or by the user, it must be clear that more information is available; 3) the choice of visible and minimised information must be based on a metric that is clear to the person using the system; 4) the system must always allow the user to foreground what attribute is important to them in their exploration [14]. The result of this design process is Continuum. In the following sections we describe the three main panels of Continuum, how it has been designed with hierarchies, relationships, and scale in mind, and key notes on implementation.

Interaction Design

Continuum, as detailed in Figure 3, has three main panels: 1) the timeline overview, 2) the timeline detail view and 3) the dimension filter. All three panels have novel attributes that have been designed to deal with the problems of scale, hierarchy and relationships described above.

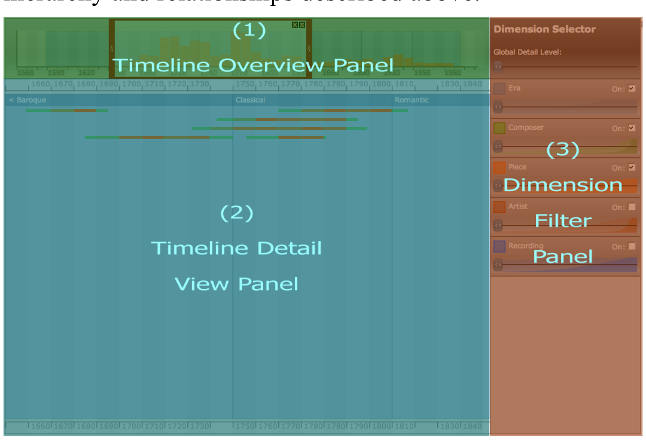

Figure 3: The three panels of Continuum.

The Overview Panel

The overview panel, top left, differs from existing timeline visualisations by presenting a scalable histogram overview. Typically, timeline visualisations that include an overview, such as the Simile timeline, simply show the same information as the detail view, but on a much smaller scale. However, for such tools, as the detail view overflows, so does the overview. At such points of overload, an overview fails to provide a complete representation of all the information that cannot be seen in the detail view.

In Continuum, the overview histogram scales to continually provide a complete representation of the whole dataset. As expected, the horizontal span of the overview panel represents and maintains the complete timespan of the dataset. However, instead of using vertical space to stack concurrent events, the Y-axis is used to quantify the focal data item of the domain. In Figure 4, the overview histogram is showing the number of classical music pieces composed in each year. The sample rate is calculated by dividing the full range of the data into the most minimal chunks that can be visualised. Thus as the width of the tool is reduced, or the temporal breadth of the dataset is increased, the span of time represented by a single bar will increase.

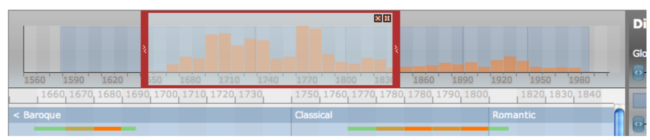

Figure 4: Continuum's overview panel displays the core data as a histogram, to scale up to large datasets.

Discussions with participants in the early stages of prototyping revealed that although different temporal metadata could be visualised in a timeline, such as composer or era, the timeline should maintain its focus on the core data item; in this case: piece. This is in line with previous research [29], which showed that although the volume of information spaces such as classical music could be quantified by any metadata (e.g. composer, instrument), in flexible semantic environments, like those often represented by faceted browsers, users primarily expected volume indicators to count the domain focus data. Subsequently, the overview histogram counts piece, and the related categorical metadata is used to cluster the focus data in the detail view.

The Detail View Panel

The detail panel shows the information bounded by the viewfinder of the overview panel. If the viewfinder is expanded, then the time spanned by the detail panel will increase. Dragging the viewfinder left and right scrolls the detail panel. Similarly, scrolling the detail panel moves the viewfinder in the overview panel. This is fairly standard behaviour [25, 12] that will be familiar to, and perhaps expected by, many users. Although other timeline visualisations [22, 25] are able to display different types of data, the hierarchical relationships between these data are often left to the user's perception of concurrency in the timeline. Colour [25], linking arrows [11], and even size [22] have been used to indicate relationships between events. However, each of these approaches have limitations in terms of scale. Colour can only be used to visualise one class of relationships at a time: either how visualised information is related to an era, or how it is related to a composer. Linking arrows can be used to convey many relationships concurrently, but can lead to a very busy interface. Multiple types of relationship may wish to be expressed, but there are limited different types of arrow. Using size is a volatile method of categorising information. Firstly, size is often associated with importance. Secondly, a flurry of 'large' events can take up a lot of screen space very quickly.

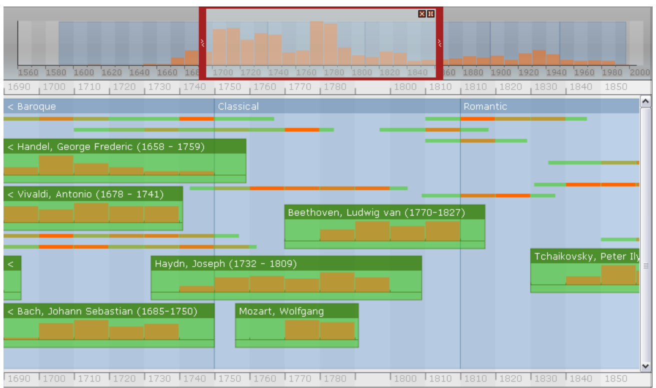

Figure 5: Child nodes are displayed within Parent nodes. Here, Composers are green boxes within the larger (background) blue Era boxes.

In Continuum, the temporal relationships of information are represented as a set of bounding boxes. In Figure 5, the Composers are visualised as spans within the larger blue spans representing Eras. Each Piece is then visualised within their Composer. Like the overview panel, histograms are used to quantify larger volumes of information that cannot be viewed in detail, such as viewing a Composer's Pieces. However, the level of detail can be controlled using the dimension filter described below.

The Dimension Filter Panel

This unique aspect of Continuum gives control of the abundance of content to the user. As displaying all the information from each dimension would overload the detail panel, the dimension filter panel allows users to control the level and type of detail displayed. Put simply, this allows the user to specify that they want to see lots of information about composers, minimal information about their compositions and absolutely no detail about later recordings. To express this, each dimension has a slider and a checkbox. The checkbox allows the user to define which dimensions are visualised at any one time. If the slider is at its leftmost point, the majority of Composers are represented by flat horizontal lines. As the slider is moved towards the right, the most prominent Composers begin to grow in height to display more detail. (The 'prominence' metric can be changed as required – for our prototype it is simply the number of compositions). As the slider approaches the right, all of the composers become expanded. By doing so, more vertical screen space may be needed, and so by expressly requesting more detail, the user is implicitly creating the need to scroll the detail panel vertically.

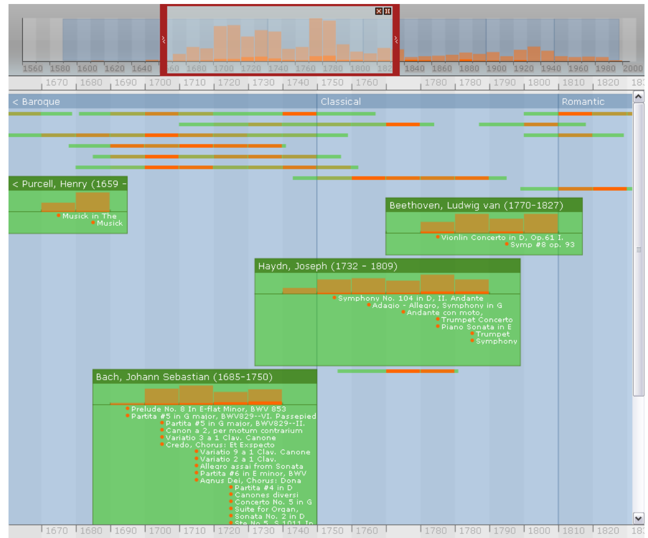

Figure 6: As the Piece slider is dragged to the right, an increasing number of pieces are visualized and labelled.

Regardless of the state of the Composer slider, the Piece slider controls the amount of information shown about individual pieces inside each expanded Composer. An overview of the pieces always remains visualised as a histogram. By moving the Piece slider to the right, the user requests more information about more of the pieces, which requires further vertical space. This information is shown by 'dropping down' or expanding pieces from the histogram, and labelling them, as in Figure 6. If the user were to move the Composer slider left again, the less prominent composers would be minimised to flat lines, so that the user would see a lot of information about Pieces, but of only the prolific Composers. Finally a global slider can be used to increase or decrease the detail level of all the active dimensions.

Designed for Hierarchies

Hierarchies have been supported here through both the detail and dimension panel. By representing child nodes as entities within parent nodes, the visualisation can drill down through many hierarchical levels to find information. As showing a full hierarchy would introduce scale problems very quickly, the dimension filter panel allows the user to specify the levels and detail that they wish to view. Further, individual entities can be pinned open so that they always shown regardless of the state of the sliders. This allows a more specific investigation into certain entities. Similarly entities can be pinned down, so that they never take up screen real estate. Era/composer/piece are categories, in Hearst's sense of categories as facets [9], and as such are flexibly associated as hierarchies in Continuum. For example, we can display era -> composer -> piece, or era -> piece. The data is not a rigid hierarchy, but a hierarchy by association. This distinction and our display of embedded entities is in contrast to say, Lifelines, which displays categories on different rows, and even with linking and colour, leaves the user to make some spatial association about relationships between Gantt-like lines.

Designed for Scale

As both the size and dimensionality of the information can lead to information overload, Continuum has been designed carefully to deal with scale. When the amount of entities would be prohibitive or uninformative to show completely (such as Piece within Composer), we show the information as a histogram. Thus, where existing tools will fail at showing complete information, Continuum conveys alternative information: relative quantity. Histograms are used in both the overview panel and within entities of the detail view.

However, as the user wishes to further investigate entities in the timeline, they can purposely expand its allocated screen real estate and see its information in more detail. By choosing to increase this size, they can also see that the detail panel has to expand vertically, but that the change is under their control. The pinning technique can be used to select particular entities and drop the detail on the rest, reducing the vertical space used; again under the user's control.

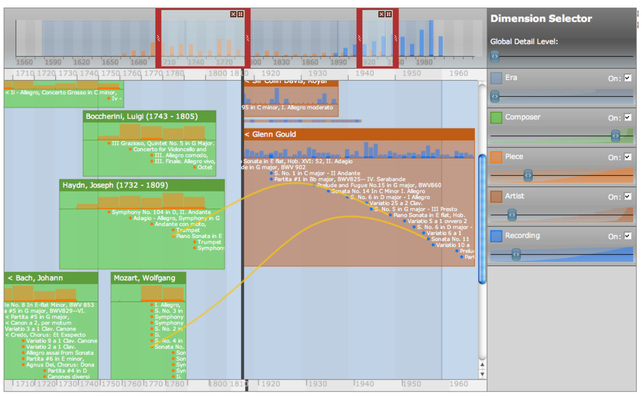

Figure 7: Non-hierarchical relationships, or those that do not occur at the same time, such as a composition and its later recordings, are displayed with arching connection lines.

Designed for Relationships

One limitation of embedding child nodes within parent nodes is that it cannot visualise relationships that are not temporally connected. Composer and composition occur in the same timespace, but the London Symphonic Orchestra's recording of Bach's Concerto No. 5 cannot be represented within each other, as they happened at separate times in history. To represent this special class of relationship, arrows have been used. Whereas SemTime [11] uses arrows for many types of relationships, here arrows are used for one class of relationship in a dataset, thus minimising its adverse effects. With the combination of hierarchical relationships represented nested spatially, as well as a non-temporal relationship by arrows, a significant amount of information can be simply conveyed. Clearly, there may be times when multiple types of relationships between non-temporal information may wish to be displayed. This is a topic for further discussion.

Viewing non-temporal relationships can be done in two ways. Such relationships are only shown for items that are currently in view. Either a user can expand the viewfinder to cover both events or they can create a second viewfinder. The two viewfinders are displayed in parallel, splitting the detail panel in half vertically, as in Figure 7. As the relative information is displayed concurrently, the lines connect the items across the two views.

Implementation

Continuum has been developed as a standalone advanced JavaScript widget that can be embedded into any webpage and fed with data from third party JavaScript widgets, or from an XML/JSON source file. It makes use of the "CANVAS" HTML object, which is supported by many modern browsers. The CANVAS tag allows for more control over the drawing surface of a webpage, and gives Continuum the ability to visualise complex relationships between temporal data.

The easiest way to inject data into Continuum is by loading data from an XML file that follows the Continuum data template. Figure 8 shows an extract from the classical music data set: data is wrapped up into two main XML elements, EventCollection and Event. The EventCollection element identifies a collection of events that are all of a particular type, with Continuum using the type attribute to define how the data is rendered. There are 3 types of EventCollection that Continuum currently supports:

SequentialRange: a collection of events that have defined start and end times that define periods in time that are sequential, although the dates of these Events can overlap. Sequential Ranges will be rendered in the Timeline Overview and Detail View as a background range, with events from other collections rendered over the top.

Range. The Range collection describes a group of events that, similar to the SequentialRange collection, have start and end times. The main difference with the Range collection is that many of the child events will have overlapping time periods, therefore they are rendered as individual ranges only within the Timeline Detail View.

Point: a collection of events that only have a single point in time. These events are rendered as a histogram in the Timeline Overview and as a combination of histogram and temporal nodes in the Main View depending on how the data hierarchy is defined.

EventCollections also have other attributes that describe them. The label attribute is used to identify the name of the collection, the on attribute indicates whether the collection is rendered by default and the startLabel/endLabel attributes are used to label temporal data associated with child events.

The Children element of the EventCollection is used to create a data hierarchy. When event collections are rendered, any Child links between dimensions are used to link events together, for example an Range event collection that is linked with a Point event collection will result in the Point collection being rendered as a histogram visualisation within the Range.

The Event element describes an individual event in temporal space, whether it is a range event with start and end dates, or a point event with a single temporal element. The main attributes of the Event element are used to describe its temporal values, assign it a unique identifier and provide a readable label.

```
<EventCollection label="Era"
                 type="SequentialRange"
                 on="true"
                 startLabel="Era begins"
                 endDate="Era ends" >
    <Children>
        <ChildCollection label="Piece" />
    </Children>
    <Event id="136"
           label="Modern"
           weight="91.0" >
        <Date start="-1893459600"
              end="-315622800" />
    </Event>
...
<EventCollection label="Composer"
                 type="Range"
                 on="true"
                 startLabel="Birth Date"
                 endLabel="Death Date" >
    <Children>
        <ChildCollection label="Piece" />
    </Children>
    <Event id="659"
           label="Mozart, Wolfgang Amadeus (1756-1791)"
           weight="117.0" >
        <Previews>
            <Preview type="image"> http://www.classical-
composers.org/img/mozartwa2.jpg
            </Preview>
        </Previews>
        <Date start="-6753200400"
              end="-5648662800" />
    </Event>
...
</EventCollection>
<EventCollection label="Piece"
                 type="Point"
                 on="true"
                 startLabel="Composed" >
    <Event id="1049"
           label="Concerto for Violoncello and Orchestra
                  in D major (G.476)-Allegro piacere"
           weight="0.467359991055366" >
        <Date start="-6122048400" />
        <Previews>
            <Preview type="audio"> http://beta.mspace.fm/n/1230.mp3
            </Preview>
        </Previews>
        <Links>
            <Link type="parent" node="583" />
            <Link type="parent" node="314" />
        </Links>
    </Event>
</EventCollection>
```

Figure 8: Example of data XML format.

Events can be given importance weightings using the weight attribute. These weightings are used in the semantic zoom. Each dimension in a data set (for example Era, Composer, Piece) is marked up as Event Collections that hold a number of Events. The weighting can be generated using some metric that specifies the importance of an event over others, such that the most important are expanded at a certain zoom level. In our example, we specified the weighting as the number of compositions of a composer, such that the most prolific composers would be expanded at lower zoom levels. Other potentially more meaningful metrics can be used. As an example, we can record the number of times events are manually expanded, and use this as a multiplier on the importance weighting of "most visited composers".

Using the Links element within an Event can be used to create relationships within the data. There are two types of Relationship that can be defined using the Link element. The first is that of a hierarchy, such that, given the level of zoom, events are rendered in boxes that represent parent events. In our Classical Music example timeline, composers are parents of compositions and are rendered as boxes that contains the composition point events. The second type of event is that of a direct relationship link between point events. In our example, a recording event is directly related to the composition event, for a particular piece of music. This allows the arrows to be rendered, as in Figure 7.

Embedding the timeline on a web page is a simple process, as seen in Figure 9, that involves simply importing the Timeline JavaScript, and initialising a Timeline object when the page loads. The Timeline object accepts one single parameter, which is the id of an HTML div element that is used by the timeline to render to the page. A method called addViewport is used to create an initial viewport into the data, using the passed start and end dates to define viewport. Finally the data can be loaded from an XML file.

Continuum provides the ability to load data from both JSON and XML. The XML file that is passed in Figure 8 is transformed into a simple JSON representation using XSLT. Using XML is the easiest way to use Continuum, but the ability to inject JSON directly allows other JavaScript widgets to dynamically add data to the timeline.

```
<head>
<script language="javascript"
        src="js/Timeline.js"
        type="text/javascript"></script>

<script language="javascript" type="text/javascript">
<!--
       function initialise()
       {
               window.timeline = new Timeline("Timeline");
               window.timeline.addViewport(
                new Date(1650, 0, 1, 0, 0, 0, 0).getTime(),
                new Date(1850, 0, 1, 0, 0, 0, 0).getTime());
               window.timeline.loadXmlFromFile("data.xml");
       }
-->
</script>
<link rel="stylesheet"
      type="text/css"
      href="css/generic.css" />
</head>

<body onload="initialise()">
    <div id="Timeline"></div>
</body>
```

Figure 9: Example of how to embed Continuum into a webpage.

EVALUATION

We describe evaluation with a second data set to demonstrate generality, and describe two user studies to evaluate our design.

Dataset Evaluation

The previous figures have used a classical music dataset. Because the Continuum prototype has been designed with generality in mind, it was an easy task to use a further dataset that has its own unique challenges for an evaluation of the system's performance: a set of NewsFilm clips spanning 100 years and categorized into dozens of Themes and hundreds of Topics. Figure 10 shows this dataset. While the categorisation itself has no specific temporal data, information may be inferred by the layout of the temporal instances within them. Interesting temporal aspects are derived from the date of the clips that were categorised in that way. On the timeline the Sport theme appears when the first clip about Sport was published and finishes when the last clip was published. Subsequently, many of the Themes would appear concurrently throughout history. Further, each clip may have been categorised in many ways. Thus, a single clip appears simultaneously in an array of concurrent Themes. These are aspects of the design that were not brought forward by the Classical Music dataset alone.

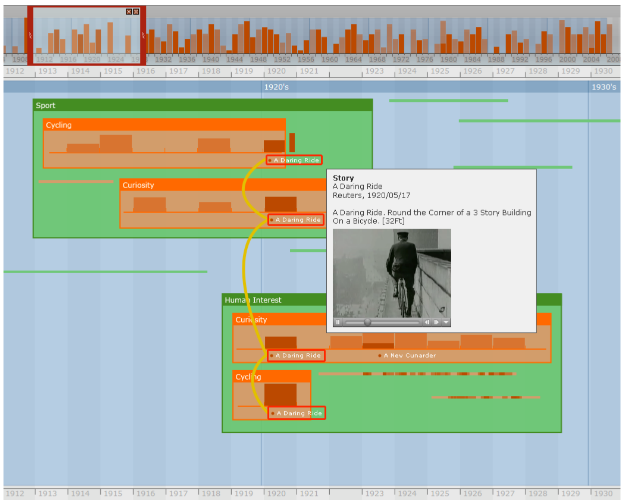

Figure 10: Continuum with a newsclips dataset.

The NewsFilm dataset (Figure 10) is an example of a much vaster scale. As mentioned previously, Simile begins to struggle when visualising more than 700 items. The classical music set was 1,500 items, and when shown in a Simile timeline, many of the datapoints are not visible unless the textsize is reduced to an unreadable size (Figure 11). Even then, there are points in time which cannot be fully represented. The NewsFilm dataset contains 11,000 items, which is further categorised into over a hundred subject areas. It is important to see that, by applying this dataset, Continuum can continue to represent thousands of items. The histogram scales well to represent all 11,000 items and the content of detail view can be controlled so that it is manageable.

Interaction Evaluation

We carried out two lightweight evaluations of our application. In each we compared Continuum with the popular Simile timeline widget via a paired think aloud protocol [28]. Comparison is an effective way to help participants articulate what they find works and does not work in a design. The think aloud protocol is likewise an effective mechanism to encourage participants to describe what they are doing and thinking about the interface as they carry out a task. In the paired protocol, one participant navigates the interface to carry out the task, the other participant drives the interface. The feedback generated between the pair in communicating what they are doing and how they need to do so is often more informative than in the single participant think aloud.

Participants were given both unstructured and structured time with the two timeline tools. Both instances contained the same classical music dataset and the tool order was alternated between pairs. In the unstructured time, we encouraged participants to explore each interface. This approach let us begin to see how one tool was accepted/perceived relative to another. The structured time involved the pairs working together to complete specific tasks with the timelines. These tasks involved both simple and complex tasks, where simple tasks that involved simply locating entities on the timeline and more complex tasks involved handling scale, hierarchy, and relationships. For example, a simple task was to 'Find Beethoven'. A complex task, for example, was to find when Beethoven began composing pieces, or to find a time when he was composing a lot or not at all. We debriefed with the participants, using semi-structured interviews about their general perceptions of the designs and their strengths and weaknesses.

Six people from our research group participated, where participants took turns driving and navigating through the interface. We performed the study using Firefox 2.0 on a 23" screen driven by a MacBook running OS X 10.4.9 with 2GB RAM and 2GHz Intel Core Duo processor. Comments noted throughout the evaluation were collated using qualitative coding methods [17].

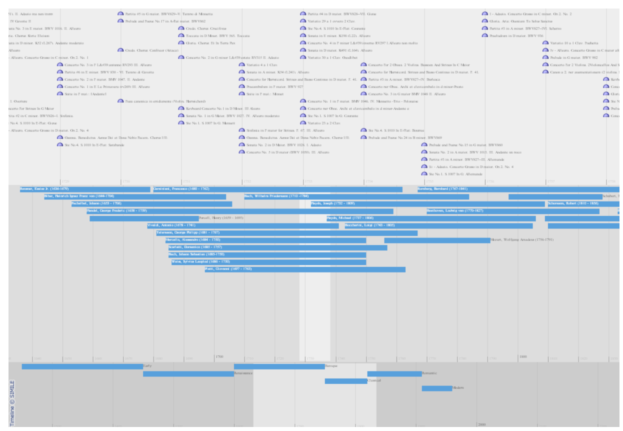

Figure 11: The Simile timeline with the classical music dataset. The text has had to be reduced to an illegible size to fit on all items.

Preliminary Results and Analysis

Our initial study results foregrounded design tweaking issues rather than fundamental functional problems in Continuum. For example, some participants found that sliders for filtering data were labelled too ambiguously to understand how they worked without having to try them first. Also, the immediate feedback (absent in Continuum) when scrolling in Simile was very popular with participants. Yet, participants commented that Continuum let them explore data in rich and innovative ways. For example, as one participant expressed when looking for all the pieces by a particular composer "[Continuum] is better because there is absolutely no confusion over who composed which pieces". Another commented "…it is intriguing to see [composer] had a dry period there". This type of visualisation was made apparent by the histograms matching compositions against period, as shown in Figures 5 and 6. All tasks were completed by participants in Continuum, though Simile was not able to support some of the more complex tasks such as relationships between composers and compositions. Some participants were particularly attracted to the control they could have over the data facets by using Continuum. Others, however, found that the Simile interface felt clearer and therefore simpler to engage, and so preferred it, even though it could not support some of the exploration tasks. This finding became a strong motivator in our refining of Continuum's design: our goal was to try to achieve as much as possible the ease of first use experience of Simile, while maintaining the exploration features in Continuum.

Second Study Results and Analysis

Based on the results of our first study, we refined the design to address in particular issues of interface clarity. This included: a) refining the slider design to give clear feedback on the degree of change that would be affected by having the slider at a given setting, b) ensuring that movement in the top overview pane was visually connected with the location in the main view, c) more prominent timescales, and d) tooltips only on hover. We invited the participants back and reran the first study protocol with the revised UI. The overall feedback was very positive, with one participant, who previously preferred Simile, stating "[this version] has definitely improved; it has addressed enough of my concerns to prefer using it over Simile now." This follow-up study backed up our previous conclusion that the concerns of users in the first study were simply aesthetics that could be readily addressed. In particular, we were able to see that the additional functionality and complexity of the UI did *not* inhibit people from using the tool. In fact, in the follow up study, participants made it clear that they would choose Continuum now *because* of the additional mechanisms for exploring temporal data it affords.

How to best set the behaviour of one control is still to be determined: the impact of the global slider on the individual sliders. The current default is that moving the global slider aligns all the facet sliders in line with that slider. Individual sliders can then be moved on their own, but moving the global slider again resets all sliders to that point. Some of us strongly felt that the global slider should only move the individual sliders relative to their current setting (like a master volume control on a mixing board with individual volume levels per track) others in the study, however, felt that the zeroing behaviour of the global slider was what "made sense." We will likely find a way to support both models for the slider. Minor aesthetic improvements as suggested by participants are also a topic for further work.

Ongoing Work

One dataset condition remains a particular challenge to represent: we call it *non-temporal multiple categorisation*, seen in the NewsFilm Online example in Figure 10. In that case facets like Theme and Subject have very amorphous temporal qualities - these themes likely have, in this case, news clips associated with the theme across broad periods. A newsclip is itself associated with several themes and subjects, as we see in the example. In our current implementation we present each Theme as a span of time, and have the same clip appear within each appropriate topic. We are not entirely persuaded that this is the optimal approach since the temporal quality of such generic themes is not necessarily particularly meaningful if a category exists in all times. In these cases, there may be richer ways either to infer or associate more meaningful kinds of categories if they are not explicitly stated in the data. In the news set for example the topic "war" exists, but not the sub topic "WWI." It would be possible to refine articles into such a subcategory that has distinct temporal points. This invites the next question, however: what is the heuristic for mapping as a temporal hierarchy the period of the actual event, vs categorizing articles reflecting on that event at a later point? In our current representations there is no way to distinguish between the event and discussions of it. For example, in our mapping of classical music, recordings of baroque music show up against the background of the Modern classical music period when the Era facet is set to On. This approach would suggest that representing an instance in a facet level within a hierarchy should only apply to the instances *of* that time rather than *about* that time. In other words WWI would appear as a temporal hierarchy within the 20th Century and only spanning those dates of the actual event. The question becomes under what category would articles about WWI appear, especially since these may appear at any point in time? We do not have a clear heuristic for this problem.

CONCLUSIONS

We have presented Continuum: a timeline visualisation tool that can be used to represent and interrogate large amounts of hierarchically-structured temporal data. The controls allow users to seamlessly move between summative and informative views by adjusting the amount of detail portrayed at each level of a hierarchy. Further interaction allows the user to choose the state of individual entities within a facet, so that they may persist despite the varying level of detail portrayed. At all levels of detail, the important information is presented and redundancy is avoided.

The proposed timeline makes several contributions to timeline visualisation work: (1) to our knowledge, Continuum is the first timeline to represent nested visualisations of temporally associated hierarchies. (2) It strongly supports user-determined information focus: by allowing the user to control detail of each facet individually, the user can get at what they want to know. (3) Support for user focus helps provide useful information even when in a fully zoomed out view: the user controls which attributes they wish to see stand out in relation to that overview mode. (4) The approach is sufficiently generic to work over a range of temporal data sets. (5) The use of histograms and detail controllers allows Continuum to visually represent and interrogate with meaningful levels of detail over significantly larger data sets than we have been able to provision in currently available services like Simile.

The ability to use direct manipulation techniques to rapidly carry out complex queries is a powerful approach for exploratory investigation. Continuum provides another mechanism for this kind of exploration of the rich associations in faceted data, in this case focusing on temporal data in particular. These types of possibilities enable different questions to be asked (as brought up in our studies): one sees a dry period: what are the causes? What else is going on at that time? In related projects, we will be carrying out longitudinal studies to see how these new modes for easily exploring temporal data will benefit scholarly research in both journalism and musicology (our test sources). We also plan to see how such temporal mappings can be integrated with faceted geospatial representations to add the where to the when and what of exploratory search visualisations.

ACKNOWLEDGEMENTS
Thanks to David F. Huynh for the informative discussions involving the Simile Timeline and related work. We also thank the participants in our studies, and the reviewers for their helpful comments.

REFERENCES

1. Apperley, MD and Tzavaras, I. and Spence, R. A Bifocal Display Technique for Data Presentation. *Proceedings of Eurographics*, 1982, pp. 27-43.
2. Artemis Views, AISC Corp, http://aisc.com/Product/2
3. Bade, R. and Schlechtweg, S. and Miksch, S. Connecting Time-Oriented Data and Information to a Coherent Interactive Visualization. *Proc. of CHI'2004.*
4. Bederson, B. B. and Hollan, J. D. 1994. Pad++: a zooming graphical interface for exploring alternate interface physics. *Proceedings of UIST*, 1994, pp. 17-26.
5. Brodbeck, D. and Girardin, L. Trend analysis in large timeseries of high-throughput screening data using a distortion-oriented lens with semantic zooming. *Proceedings of InfoViz*, 2003.
6. Chittaro, L., Combi, C., Representing of Temporal Intervals and Relations: Information Visualization As-pects and Their Evaluation, *Proc. of TIME*, 2001, p.13.
7. Cousins, S., Kahn, M., The visual display of temporal information, *Artificial Intelligence in Medicine 3(6),* 1991, pp. 341-357.
8. Furnas, G.W. Generalized fisheye views, *Proceedings of CHI*, 1986. pp.16-23.
9. Hearst, M. Next Generation Web Search: Setting Our Sites. *IEEE Data Engineering Bulletin, Special issue on Next Generation Web Search, 23(3)*:38-48, 2000.
10. ILOG Gantt, ILOG, http://ilog.com/products/ganttnet/
11. Jensen, M. Visualizing Complex Semantic Time-lines. NewsBlip Technical Report NBTR2003-001, http://newsblip.com. 2003.
12. Jog, N., Shneiderman, B. Starfield information visualization with interactive smooth zooming. *Proc. of IFIP 2.6 Visual Databases Systems*. 1995, pp.1-10.
13. Koike, Y., Sugiura, A., & Koseki, Y. TimeSlider: An Interface to Time Point. *Proc. of UIST*, 1997, pp. 43-44.
14. Kules, B., Kustanowitz, J., and Shneiderman, B. Categorizing web search results into meaningful and stable categories using fast-feature techniques. *Proc. of JCDL '06*.
15. Kumar, V., Furuta, R., and Allen, R. B. Metadata visualisation for digital libraries: interactive timeline editing and review. *Proceedings of DL* . 1998 pp.126-133.
16. Mackinlay, J. D., Robertson, G. G., and Card, S. K. The perspective wall: detail and context smoothly integrated. *Proceedings of CHI*. 1991. pp.173-176.
17. MacQueen, K., E. McLellan, K. Kay, and B. Milstein. Code book development for team- based qualitative analysis. *Cultural Anthropology Methods Journal 10 (2)*: 1998. pp.31–36.
18. MinuteMan, MinuteMan Systems, http://minuteman-systems.com/minutemanproductdetails.htm
19. MS Project, Microsoft, http://office.microsoft.com/project
20. OmniPlan, The Omni Group, http://omnigroup.com/applications/omniplan/
21. Plaisant, C. and Mushlin, R. and Snyder, A. and Li, J. and Heller, D. and Shneiderman, B. LifeLines: Using Visualization to Enhance Navigation and Analysis of Patient Records. *Proceedings of AMAI*, 1998, pp.76-80.
22. Plaisant, C., Milash, B., Rose, A., Widoff, S., and Shneiderman, B. LifeLines: visualising personal histories. In *Proceedings of CHI*, 1996, p. 221-227
23. Richter, H., Brotherton, J., Abowd, G.D., Truong, K. A Multi-Scale Timeline Slider for Stream Visualization and Control, GVU Center, Georgia Institute of Technology, Technical Report GIT-GVU-99-30. 1999. http://hdl.handle.net/1853/3397
24. schraefel, m.c., Wilson, M., Russell, A., and Smith, D. A. mSpace: improving information access to mul-timedia domains with multimodal exploratory search. *Commun. ACM 49, 4* (Apr. 2006), pp.47-49.
25. Simile Timeline: http://simile.mit.edu/timeline/
26. Tufte, E. R. *The Visual Display of Quantitative Information*. Graphics Press. Cheshire, Connecticut. 1986.
27. Tufte, E. R., and contributors. "Project Management Graphics". Ongoing forum discussion, http://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=000076
28. Wildman, D. 1995. Getting the most from paired-user testing. *interactions 2,(3)* (Jul. 1995), 21-27.
29. Wilson, M. L. and schraefel, m.c. mSpace: What do Numbers and Totals Mean in a Flexible Semantic Browser. *Proceedings of International Semantic Web User Interaction Workshop at ISWC2006.*