Automated Capture and Representation of Date/Time to Support Intelligence Analysis

David M. Cassel
Lockheed Martin
P.O. Box 8048
Philadelphia, PA 19101
610-354-4909
david.cassel@lmco.com

Sarah M. Taylor
Lockheed Martin
4350 N. Fairfax Drive
Arlington, VA 22203
703-351-4440x135
sarah.m.taylor@lmco.com

Gary J. Katz
Lockheed Martin
P.O. Box 8048
Philadelphia, PA 19101
610-354-5880
gary.j.katz@lmco.com

Lois C. Childs
Lockheed Martin
P.O. Box 8048
Philadelphia, PA 19101
610-354-5816
lois.childs@lmco.com

Raymond D. Rimey
Lockheed Martin
P.O. Box 277004
Mail Stop DC3535
Littleton, CO 80127
303-977-4811
raymond.d.rimey@lmco.com

ABSTRACT
Most intelligence analysis tasks require the analyst to have a mastery of a sequence of events. Tools supporting intelligence analysis must aid the analyst in untangling a variety of time related problems, including overlapping durations for events, and imprecise, incomplete, or conflicting information. Tools require both strong date and time identification/extraction and a flexible presentation of information on a time scale. We propose a method of presenting events, linked to a timeline, which addresses the problems of overlapping events, vague date/time references, and the need for the analyst to move easily between different time scales.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *graphical user interfaces, natural language.*

General Terms
Design, Human Factors, Languages

Keywords
Event analysis, intelligence analysis, time expressions, timeline

1. INTRODUCTION
Intelligence problems come in great variety, ranging from the development of situation awareness in preparation for a battle, to trying to understand the effects of a natural disaster on an economy, to following the progress and associations of a target person or object of interest. In almost all cases, an understanding of past events, and an ability to track the progress of events as they unfold, is a necessary part of the analysis applied to the problem. Additionally, fusion of information from multiple sources requires the best possible information about the time, as well as location, of the information – both the time and place of collection, and the time and place of the recorded events. Some sources of information may come to the analyst already supplied with precise date and time information of this type. However, in text sources, with which we primarily are concerned for this discussion, much information about the timing of events is vague or otherwise incomplete. Yet, the analyst must make the best of the information he has. His tools should support him.

A tried and true method for fusing and understanding events has been the timeline, a linear scale to which the analyst can link events. These were hand drawn for many years, of course, and now can be provided, in a variety of flavors, by commercially available software tools, or constructed manually (and thus tailored to a particular problem and data) using drawing software. However, there remain some challenges for automated timelines which have not been overcome to our satisfaction: automated input of information to the timeline; the effective presentation of the time scale, with events attached, at different levels of granularity; the effective representation of events with overlapping durations, as well as the ability to include events with vague date/time references and differing degrees of uncertainty. These representation issues need to be addressed with methods that are reasonably intuitive and provide clarity, not visual clutter.

We describe here our understanding of the needs that a timeline tool must meet, our approach to the automated extraction of time information from text for ingest into a tool, and our concepts for representing time information on a zoomable timeline. This project is on-going. The required information extraction capability, from text, is available from Lockheed Martin's AeroText™ software and the zoomable timeline has been implemented, with many of the features described here, in a research prototype.

2. TASKS REQUIRING TIMELINES
Our timeline tool targets event information included within text. The types of text for which it may be useful are news reports, government message traffic, and internet sources. Because of the volumes of such materials and the large numbers of events

involved, particularly at the single event level, tools to automatically assist intelligence analysts in discovering and understanding events of interest are becoming increasingly important. We start with some examples of the types of analysis for which timelines can be particularly helpful. These are provided to clarify the need for a tool for those readers not familiar with the methods of intelligence analysis.

Recurring patterns of events can suggest methods of operation for planning and executing terrorist acts, for money laundering, drug smuggling, and any number of criminal activities. They can be helpful in understanding negotiating behaviors and diplomatic maneuvering as well as military operations. What series of observable behaviors typically appear before elections in this country? What are the typical preparatory steps for a certain organization to carry out a terrorist attack? Timelines make such patterns easier to see and to compare.

Exact sequences of events can be highly important. Did a certain meeting take place before or after the decision was made to carry out a particular policy? Are policy statements regularly softened or hardened after input by any one official? Did a certain piece of information become available only after the press conference by a particular agency? This kind of sequence information can be critical to building an analyst's understanding of the viewpoints and information that influenced policy determinations or decisions to act. Using a timeline can help the analyst determine and explain such sequences. [4]

Timelines are particularly useful in deconflicting different versions of events. If incompatible time expressions are used, it can be difficult to see the conflict between different versions of an event until the time expressions have been normalized – one of the effects of putting events onto the timeline. For example, the conflict between the following two statements is more easily spotted in Figure 1 than in the text versions. "On March 15th (2001 understood from previous context), Mr. Jones left London for Kabul, beginning a trip of several months to countries in South Asia and the Arabian Peninsula." "Mr. Smith arrived in London in 2000, establishing a close working relationship with Mr. Jones there during the late spring of the following year." Mapping the events of both statements to a single timeline shows immediately that the statements cannot both be true.

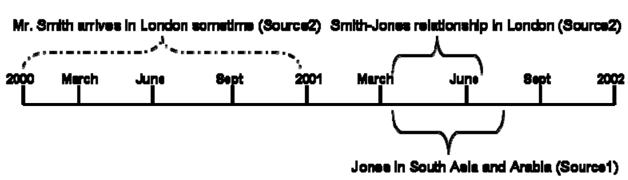

Figure 1. Depicting events on a timeline aids deconfliction.

Plaisant et al. showed in [5] the flexible utility of timelines by applying very similar systems to the juvenile justice and medical domains. For juvenile justice, the timelines illustrated events in from the records of troubled youths, enabling an easier method to search for patterns of behavior. In the medical domain, the LifeLines tool showed problems, allergies, diagnoses, complaints, lab work, imaging, medications, immunizations, and patient communication on a single view, while allowing the viewer to specify the interval and level of detail to be observed.

3. REQUIREMENTS FOR TIMELINES

Guidelines for annotation of time expressions in text [6], [3] make clear the wide range of time expressions which must be captured and represented by timelines. These include not only the obvious, such as "yesterday", "10:50am", "following", and the information conveyed by verb tenses, but also the less overt, such as the existence of states, e.g. "married" which in general usage imply a previous event, e.g. "wedding", or states of planning or expectation, which mean that something has not yet occurred, but may do so in the future. To truly capture the useful information from the text, any one of these kinds of time expressions must be able to be represented on a timeline.

There is a legitimate question, however, whether the timeline should in fact be made to represent every type of time expression; is it useful to try to recreate in the graphic form all the nuances of the linguistic form of the information? Without considerable experimentation and testing, we do not believe that question is definitively answerable. However, the approach we take is that there is a set of features that must be represented on a timeline, to which most time expressions can be mapped, and that these cover most of the cases for which timelines are typically used. We begin by addressing these clearly necessary features with our timeline tool.

Timelines are used to represent the ordering and spacing of events, that is, something, signaled by a verb in the language, that happened, a change from one state to another. States of being can also be represented, e.g. "Mr. Smith was in London from May until September", but these may be considered in terms of their beginning and ending events, e.g. in this case, possibly an arrival in May and a departure in September, depending upon the context. But every event, or state, that must be represented is either a point or an interval in time. The differences in time expressions reflect not a wide range of different kinds of time, but differing degrees of knowledge and precision of expression concerning the point or interval of time of the event. If we wish to be extremely accurate, of course, all events are intervals of time, but many of them are short enough, in whatever time scale is being used, to be usefully represented as points.

Vague or missing time information is extremely common in text sources. Information may simply not be known, or even not knowable, in any greater detail or with any greater level of confidence than is expressed in the source. Where an event has not been observed directly, but is being inferred from the existence of other events, or is known from reporting by a third party, the language may be a faithful expression of the level of detail available to the speaker. In other cases, the level of detail available in the text is related to the importance accorded to the event in that particular discussion, with the key events of a news report, for example, more likely to be specified closely and the background information referred to with more general expressions of time. Thus, the same event may be referenced in one source with a specific time expression and in another with something much vaguer. These imprecise expressions – "sometime last month", "several times a year" – may be all that an analyst has available, and must be represented as faithfully as possible on the timeline in conjunction with the more specific information.

Reliability of the source information is also important for the timeline, as in any other analysis tool. Combined into a calculation of reliability are estimates of the worth of the original report and of the chain of automated and human processing the information underwent before it reached the timeline display. Methods for automatically deriving this kind of provenance are not yet mature enough to include in our current version of the tool. Eventually, source reliability will necessitate a second layer of uncertainty information being incorporated into the timeline, over and above the information about the imprecision of the language content which we are incorporating in this version.

Aside from imprecision of expression and estimates of reliability/uncertainty, three major challenges for representation on timelines remain: a wide range of scales, large numbers of events, and characterizing event content.

Time scales must be able to represent scales from the level of seconds to the level of centuries. Similar to the changes in scale required for geographic analysis, it may be necessary to view events at any number of levels – at a higher level to see the evolution of a situation over a ten year period, yet also being able to focus on the detailed happenings of any particular week or day within that period. This requires not only an ability to grow or shrink the scale of the timeline, while retaining the user's focus, and orientation, but also the ability to meaningfully summarize events for display on the higher scales, and to decompose those summaries into their constituent pieces for more detailed analysis. Otherwise the more general scale – the decade view for example – simply becomes indecipherable for all the detailed events that have been attached to it. So, concomitant with our work on the multiple scales of the timeline itself, is work, not reported in detail here, on the association of component events into larger and more abstract constructs.

Large numbers of events on a timeline present similar problems to other analysis tools. But the possible solutions are more constrained. Unlike a link chart, for example, the time scale, which must remain easily understood, can be less easily stretched or rearranged to accommodate more material.

Finally, for the timeline to be most useful, the events displayed on it must be represented in such a way that their most important characteristics are readily recalled and understood by the viewer. Lines, bars and dots are insufficient without additional information. It is necessary for these to be as efficient in their use of space, as intuitive, and as distinctive as possible. We are experimenting with different representations and expect to report that work elsewhere.

4. EXTRACTION OF DATES/TIMES FROM TEXT

Any system extracting date and time information in text must accomplish three tasks: first, it must locate time expressions in the text; second, it must normalize the expressions into some standard and structured format; and finally, it must associate the date/time structures with the events or relationships they describe. The system we use for date/time extraction is Lockheed Martin's Information Extraction software tool, called AeroText™. We describe our approach to the location and normalization of date/time expressions below.

Two widely recognized schema for structuring date/time information are Timex2 and TimeML. Timex2 is an SGML-based format used to represent a broad array of times, including specific dates, durations, sets, fuzzy dates, and non-specific dates [3]. TimeML includes Timex3 tags, which supplement the attributes of Timex2, as well as other tags for marking events [6].

These standards have the flexibility to represent the breadth of temporal expressions necessary for dealing with natural language. Many programming languages provide some sort of date object, but these can typically only express specific dates. In contrast, consider the Timex2 specification for three expressions: "8:15 tonight", "Thursdays in May", and "the past three summers" (these examples each assume a reference date of June 1st, 2006).

<timex2 val="2006-06-01T20:15">8:15 tonight </timex2>

<timex2 set="YES" val="2006-05-WXX-4>Thursdays in

<timex2 val=2006-05">May</timex2></timex2>

<timex2 val="P3SU" anchor_dir="BEFORE" anchor_val="2006-06-01">the past three summers</timex2>

The first example shows a specific date and time, which Timex2 represents precisely using the same level of granularity provided by the expression. The second shows a set of dates. This set is precise, but would typically require one date object per Thursday to represent it. The third example shows a case that is vague – it does not specify when the summers began or ended.

Note the difference between the "val" attributes for "8:15 tonight" and "May". In the first case, the val attribute is specified down to the minute, whereas the second drops everything after the month. This approach does not assign arbitrary values for information that is not available; rather, the representation reflects all the information given and no other.

In 2004, a Time Expression Recognition and Normalization (TERN) [7] evaluation was first conducted by a team consisting of MITRE Corporation, SPAWAR Systems Center and the National Institute of Standards (NIST). Then, in 2005, the Automated Content Extraction (ACE) [1] evaluation included TERN as a separate task as well as requiring that extracted events, in other tasks, have a time associated with them. ACE and TERN rely on the Timex2 standard.

Lockheed Martin used the AeroText tool to address tasks in both the 2004 TERN and the 2005 ACE evaluations. AeroText software recognizes temporal expressions with a hand-crafted set of rules, normalizing them for internal storage to an interval form using Coordinated Universal Time (UTC). Approximately 250 rules were required for the TERN evaluations, including adaptations of the AeroText native interval forms to the output normalized forms required by TERN.

A general issue for time handling is the interpretation of expressions with respect to a reference time, which must be identified. The reference time may be the date of the document, as in "yesterday" used in a news story, and referring to the day before the date of the by-line. However, other expressions can require more complicated reasoning. For example, the interpretation of "in July", from a document written in August, might depend upon whether an associated verb was in the past or present tense to distinguish between the preceding or the following July. Embedded time expressions, such as "on Friday night in the Fall of 1998", are another complication addressed by the system. To handle these more difficult cases, our system looks for combinations of time expressions that were meaningful as a whole and applies the rules after the initial recognition phase has identified the single time expressions.

For TERN, it was necessary to find and output times in eight categories of normalized value: calendar point (a specific date or time, such as 1999), week point (includes a week indicator such as week 20 in the year 1999), non-specific (expressions underspecified in relation to the standard), duration (fully specified time periods), token (a reference to a previously specified value); prefixed (a year plus a two character prefix), no val (the actual date value is not expressed, e.g. "the anniversary"), and other (other partially specified expressions, e.g. "tonight"). It is instructive that this kind of normalization is by itself not sufficient for representation on a timeline, but requires further interpretation. The AeroText system handled calendar point, week point, token, and prefix expressions with fairly straight-forward translations from the original system. The remaining categories, dealing with less specific expressions required more significant addition of rules.

In 2004, the TERN evaluation used two scoring programs, one from MITRE and one from NIST. The MITRE scores are similar to those used in the Message Understanding Conferences (MUC) of the 1990s. The 2005 evaluation dropped the MITRE scoring; however, we provide those unofficial scores for comparison. The official NIST scores from the 2005 evaluation have been posted on the Internet [1]. There were only four participants in 2005 TERN. This is the first year such scores have been posted publicly.

The AeroText system for TERN and ACE, called HyperTERN, achieved excellent performance in the 2004 evaluation and saw only a slight decline in 2005 due mainly to two easily fixed glitches. The VAL attribute score fell somewhat because the system misidentified the document date in two document types, causing a miscalculation of normalization values. Secondly, the function to use the document date as the default Anchor_Val failed, resulting in a drop in recall from 75% to 6% in that slot. The MITRE score shows a 15% drop in F-Measure from 2004 to 2005. The drop in NIST value score was more dramatic, going from 78.1 to 56.2. This may be due to weighting factors applied by the NIST scoring program. A full description of the NIST metric can be found with the official posting of the 2005 results. Fixing the one document date glitch allowed our NIST score to rise to 83.4, an unofficial score, but one that clearly shows a continued excellence in extracting and normalizing temporal expressions.

Despite the extra work of normalizing the extracted expressions, HyperTERN achieves processing speeds greater than 200 Mbytes per hour. This system provides the basis of the date/time identification for our timeline solution.

5. RELATED WORK

A few software tools are available addressing the general problem of displaying temporal information. However, none appears suitable for the kind of detailed event analysis that analysts must often perform. The InXight Time Wall, shown in Figure 2, presents icons and text in horizontal lanes, allowing multiple timelines, used for different series of related events, to be presented simultaneously and compared. However, since the Time Wall displays events as points in time, it is difficult to detect when events, within one series, are overlapping or simultaneous. The use of perspective to indicate earlier and later times, with the time period under study in the foreground, is more impressionistic than useful, as the events on the perspective portion of the wall are not recognizable. Visual Analytics offers VisuaLinks displaying either icons or text as a variant of their link analysis software. This tool only represents points in time with no easy way to display overlaps in related events. I2 Analyst Notebook simply allows the analyst to build his own display of a set of sequential links between events, with no time scale provided. None of these tools provides robust time processing to help ingest event information automatically. None provide features we are beginning to see as necessary to the use of timelines: representation of the uncertainty of many time expressions, rapid navigation across long time ranges, and the ability to view time sequences in both a macro and micro context.

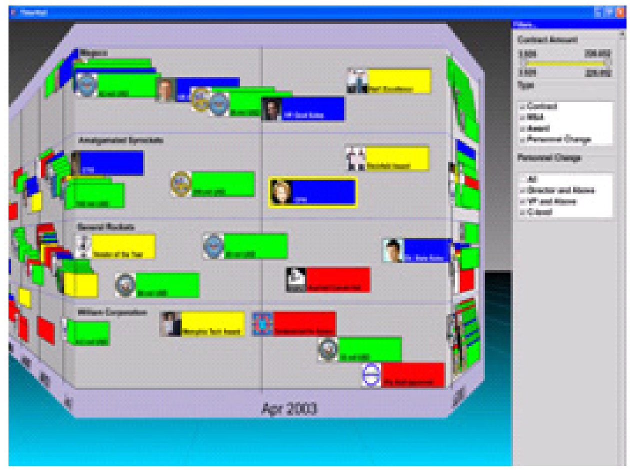

Figure 2 – InXight Time Wall

6. A SOLUTION FOR AN ANALYST TIMELINE

Effective analysis requires that time expressions be normalized and attached to events. Furthermore, effective visualization techniques are required for analysts to be able to use the information. One solution is a timeline so that the analyst can see the ordering of events and look for patterns over time. We are implementing a new timeline view that we believe will improve on previous event visualizations for analysis. Our timeline provides smooth zooming, displays fuzzy date/time expressions, provides for abstract representations, shows a context bar, and allows for stacked visualizations, as explained below. We believe these meet the requirements already detailed, for representing imprecise times, expanding and compacting time scales, displaying large numbers of events, summarizing and decomposing events, and intuitively representing the key elements of events. We will continue to test and evolve these elements of the solution.

6.1 Smooth Zooming

Time sequencing can occur at many levels of detail. Our timeline provides a smooth zoom, whereby the amount of time displayed expands and contracts in response to mouse actions. As the timeline zooms in, additional detail becomes visible – first years, then months, then days; zooming as far as seconds is possible. Events on the timeline spread out and expand their time markings in real time as the zooming continues. Zooming out brings items together and hides time markings as they become too small to be useful. This allows analysts to work at the level of detail appropriate to their tasks.

As events are drawn together, it becomes necessary to use some of the display's vertical space to show events. Boxes pile in order to maintain their correct horizontal placement. Animation is used when a box is repositioned vertically; it slides into its new location in order to help the analyst maintain orientation.

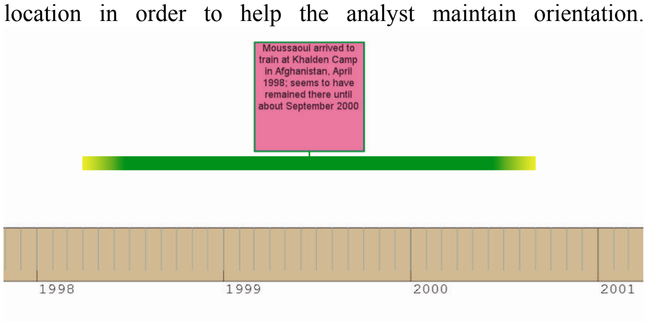

Figure 3. Gradient shading illustrates the likely range of an event.

6.2 Fuzzy Dates and Times

In order for analysts to work at different levels of detail they must understand the level of precision of their data. Improper analysis could occur if an analyst were sequencing events that occurred within minutes and the precision of the data was not easily recognizable. For example an event might start on a timeline at 1/5/06 at 12am. The display must clearly indicate whether the event began at exactly at 12:00:00 am, during the hour of 12 am, or sometime during the course of that day. This careful reflection in the timeline display of the imprecise language is necessary to facilitate correct analysis.

Fuzzy times, along with more complex time expressions (examples above), can be represented internally by the Timex2 representation. Our timeline provides a visual representation of a variety of time expressions using the granularity that is supported by the available data. We are experimenting with various gradient representations for imprecise expressions as shown in Figure 3. The precision of an event is shown by shading its timeline to describe the probability of the event occurring during that slice of the time-span. In this case, the text says that "Moussaoui arrived to train at Khalden Camp in Afghanistan, April 1998; seems to have remained there until about September 2000." The bar under the box containing this text is a dark green for most of the period from April 1998 to September 2000. However, at the beginning and end of the period, the color fades and turns yellow at the outer points. This shows that the information about Moussaoui's arrival and departure are approximate. Traditional representations would simply show Moussaoui in Khalden Camp from April 1st, 1998 to September 30th, 2000, creating the illusion that he could not have been somewhere else in the beginning or ending period. With the gradient representation, an analyst can reconcile another report that showed the subject in another location on April 3rd, 1998, for instance.

6.3 Abstraction

Each event shown on the timeline may be a single event or a collection of events. It is represented by a box depicting its contents either pictorially or in text. Double-clicking on a collection event causes it to explode to its constituent parts, as shown in the transition from part A of Figure 4 to part B. Similarly, a user may collapse the constituent pieces back into their container, cleaning up the display. A sub-item may be locked so that it remains visible when its siblings are collapsed. These transitions are animated in order to make the connection between parent and child nodes clear, and to help the viewer track the new items as the contents of the display are rearranged.

As the parts of a collected event separate, each becomes an item with its own pictorial or text representation and its own time indicator. Collection items may in turn hold other collection items. In this way, an analyst may get an overview of a set of events and then drill down to view more detail. To reduce clutter, the analyst can hide items that are not of interest. The collection items form a hierarchy, the creation and maintenance of which are beyond the scope of this paper.

As events are combined in a hierarchy, more complex shaded timelines must be created from sub-event timelines being added together. When the collection item is broken down, the shaded time display itself breaks into constituent pieces, with each box showing its own time independently. This effect can be seen in parts A and B of Figure 4. Note that in part A, a section of the line is yellow, stretching from the 11th to the 13th. In part B, we see that this corresponds to a period not covered by any of the sub-events. The color coding provides insight into the abstraction presented in part A. Similarly, the lines under the boxes in part B use gradient shading to signify the vagueness of the endpoints. This shows the uncertainty as to when events actually happened.

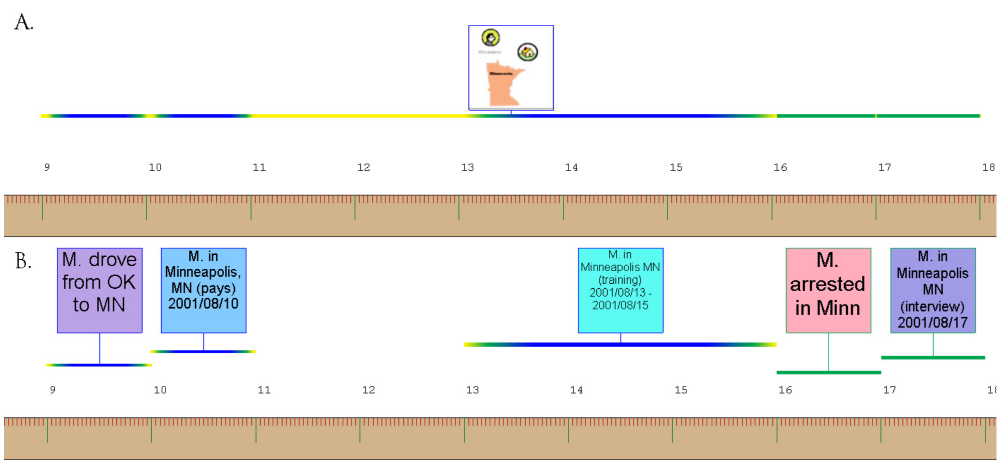

Figure 4. Abstract events contain more detailed sub-events.

6.4 Context Bar

Our timeline also employs a Context Bar, as shown in Figure 5. This bar appears just below the timeline and acts as a master timeline. It spans the range of time over which items of interest occur. Lines within the bar represent times when items of interest appear, giving the analyst context at a glance. The bar functions as a scroll bar across time: the span being displayed corresponds to the scroll bar thumb. Moving the thumb translates the view right (ahead in time) or left (backward in time). An analyst can also resize the thumb by positioning the mouse over the left or right edge and dragging. Resizing the thumb is one means of zooming the view. An approach like this was also used in the University of Maryland's LifeLines system. [5], [8]

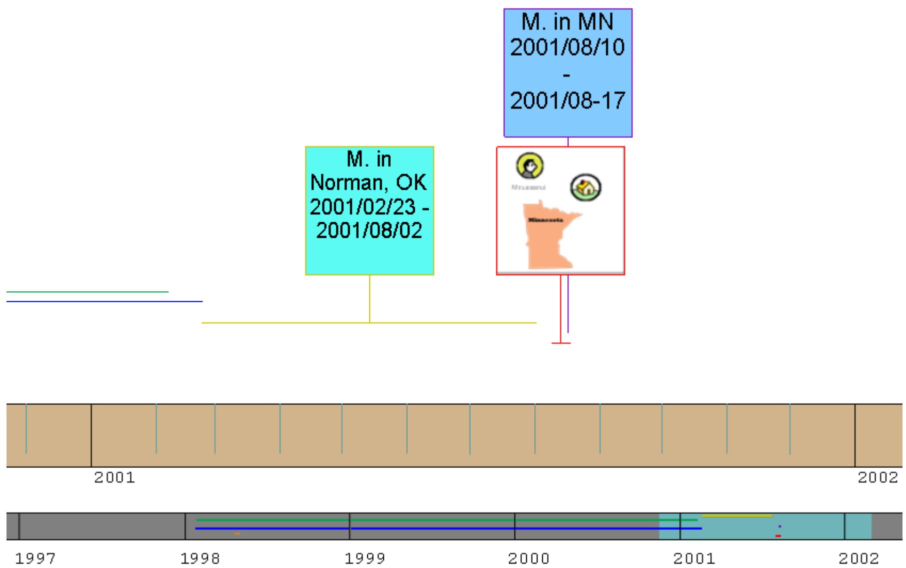

Figure 5. The Context Bar provides a broader view and zooming control.

The end points of the Context Bar are selected based on the overall span of events in which the analyst has indicated interest. In some cases, this selection can be problematic, such as "Up until 2002 …." but we use the earliest and latest events that have at least approximate start and end points, respectively.

Within the Context Bar itself, the lines indicating the presence of viewable events use the same level of abstraction as the view above on the main timeline; thus opening a container or collapsing children back into one changes the lines held within the Context Bar.

6.5 Stacked Timelines

The Context Bar provides another benefit: the system can present a stacked set of context bar timelines, with each showing a fixed period of time: a week, a month, or a year. In this way, the analyst can look for events that appear to occur at somewhat regular intervals. While data mining techniques can identify an event that occurs every Friday at 3pm, this view enables the analyst to find events that happen at less precise intervals, for instance, within the last few days of each month.

6.6 Traditional Timelines

[2] lists six shortcomings of traditional timeline visualizations.

1. *Views are discrete and fixed, thus they are bound to one specific [level of detail] and have a static size.*

In a display with a fixed level of detail, the user lacks the flexibility to determine how much information to view. Some kinds of analysis track events that happen over decades; for others, weeks or days may cover all the relevant events. Even within one of these types, a user may wish to shift back and forth between the big picture and other views of varying levels of detail.

Our timeline provides a smooth zooming capability, in which the timeline expands and contracts in response to user commands. The events on the timeline spread out or move together as the scope of the timeline changes. The context bar indicates how the period in the current view relates to the task as a whole.

2. *High-level views hide too much data, whereas detailed views suffer from lack of context and orientation.*

This is a continuation of the problem mentioned above. Neither a high-level nor a low-level view is correct all the time, so zooming plays a role in the solution. However, zooming by itself does not provide context or orientation. Our timeline provides two mechanisms: abstraction and the context bar.

The abstraction concept provides the analyst with control over the level of detail. By providing the option to limit the display to high-level concepts and then to drill down, the timeline offers conceptual context. Temporal context is provided by the context bar feature, which illustrates what portion of the overall period of interest is being shown. Smooth transitions, including animation as boxes move in the display, show the analyst what is changing as it happens. This helps the analyst maintain orientation.

3. *Scalability, e.g. for mobile devices is not supported.*

Some tools are built with the goal of scalability in mind, to allow for the same tools to be used regardless of the system on which the work is being done.

In our tool, the amount of the timeline shown depends on the width of the display window and the degree of zooming selected by the user. Given the nature of the analysis task, we believe that scalability up (for instance, to multiple screens) is more important for our work than scalability down (to mobile devices). The windows of the timeline can be expanded to take advantage of large or multiple screens.

4. *Missing support for zooming lenses, i.e. high-detail views or focus areas within a coarser time view.*

This weakness addresses the situation where groups of events are clustered at different timeline points, not distributed evenly along the line. For instance, consider the case where several events happen in January and several more happen in August. If the timeline scale is set broad enough to include all the events, very little detail will be discernible. If the scale shows enough detail, not all the events will fit on the display at once.

To address this problem, we began with the notion of a curved timeline, in which concave areas (bulging down) would bring events closer together and convex areas (bulging up) would spread them out. However, we now prefer [2]'s solution to this problem and plan to evaluate it more thoroughly. It changes the scale of some portion of the timeline to provide more detail than is shown in neighboring sections, while maintaining a straight timeline. We believe this approach provides the same ability to collapse gaps in time in a manner that is easier to understand than the curved approach we originally considered.

5. *Requirement of additional cognitive efforts for reinterpretation and orientation due to missing smooth transition between views.*

In some timelines, changing the zoom is a step function, causing a sudden change in the period of time displayed. When this happens, the user must pause for reorientation to the new display. This problem is solved in our timeline through the use of smooth zooming, in which the display gradually changes in real-time in response to the user's actions. In this way, the user is able to watch the changes as they occur and thus maintain orientation.

6. *Display of absolute time is missing, i.e. scroll-bar based views (e.g. in e-mail applications) only show relative position within the collection.*

When a timeline shows only relative position, the user can easily lose track of when the observed events actually took place. Alternatively, the user may have to refer to other data points to determine temporal position, creating a distraction and breaking the user's focus.

Our timeline uses labeled hash marks to indicate the absolute time. These hash marks spread out and new ones appear as the user zooms in. Figure 4 shows hash marks indicating days of a month, whereas the marks in Figure 5 show years. Note that in Figure 5, hash marks for the months are also visible. These marks are unlabeled due to the level of detail selected. If zooming continued, the months would be labeled once they had spread out and marks for the days would appear.

In addition to the labeled hash marks, the context bar keeps the display grounded in the overall time span of the analysis.

7. EVALUATION

We are currently working with analysts to evaluate the timeline concept and our implementation. For the first evaluation, five analysts were provided with the timeline software, two tasks and data for use in these tasks. Both tasks required detailing a chronology of attacks in the Israeli conflicts with Lebanon and in Gaza. Initial reactions indicated that the timeline concept was useful enough that we intend to proceed by addressing ease-of-use issues and continue to test. Our next test will employ a somewhat more complex analysis task, involving issues like conflicting and missing data, and multiple interpretations. In follow-on work, when the software is easier to use, we will do quantitative testing for speed and completeness of solving a problem with and without the use of the timeline.

8. CONCLUSIONS

We have indicated some of the intelligence problems for which timelines are useful to the analyst and the requirements for timelines that derive from these. We have outlined the approach we take to automated extraction of dates/times for the attachment of events to timelines. We have described the features of our timeline tool which we are implementing to address the requirements and compared our solutions to the critique of timelines contained in [2]. We believe we have addressed the major issues with workable solutions and will continue to test and refine our approach through work with real data and with the help of feedback from working analysts.

9. ACKNOWLEDGMENTS

This research has been conducted using Lockheed Martin Internal Research and Development funding.

10. REFERENCES

[1] ACE2005: http://www.nist.gov/speech/tests/ace/ace05/doc/ace05eval_official_results_20060110.htm

[2] Dachselt, R. and Weiland, M. 2006. TimeZoom: a flexible detail and context timeline. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems* (Montréal, Québec, Canada, April 22 - 27, 2006). CHI '06. ACM Press, New York, NY, 682-687. DOI= http://doi.acm.org/10.1145/1125451.1125590

[3] Ferro, Lisa. 2004. TIDES: 2003 Standard for the Annotation of Temporal Expressions. http://timex2.mitre.org/annotation_guidelines/2003_timex2_standard_v1_3.pdf

[4] Moore, David T. and Krizan, Lisa 2003. Core Competencies for Intelligence Analysis at the National Security Agency. In Swenson, ed., *Bringing Intelligence About*. Joint Military Intelligence College, May 2003.

[5] Plaisant, C., Mushlin, R., Snyder, A., Li, J., Heller, D., Shneiderman, B. (1998) **LifeLines: Using Visualization to Enhance Navigation and Analysis of Patient Records** Revised version in *1998 American Medical Informatic Association Annual Fall Symposium (Orlando, Nov. 9-11, 1998)*, p. 76-80, AMIA, Bethesda MD. HCIL-98-08, CS-TR-3943, UMIACS-TR-98-56

[6] Saurí, Roser, Jessica Littman, Bob Knippen, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky, 2004. TimeML Annotation Guidelines, Version 1.1. April 2, 2004.

[7] TERN 2004: http://timex2.mitre.org/tern.html

[8] University of Maryland's LifeLines on-line demonstration http://www.cs.umd.edu/hcil/lifelines/latestdemo/chi.html