

# TimeMerger - Disparate Temporal Data Analysis

Sally Divita  
University of Maryland  
Template Software  
sdivita@umd.edu

Martin Stolen  
University of Maryland  
Space Systems Laboratory  
mfstolen@ssl.umd.edu

## ABSTRACT

In the University of Maryland's Space Systems Laboratory, like many other such laboratories, it is often difficult to analyze the large volume of temporal data of various types created when operating robotic systems. This is due to the sheer amount of data, the differing types of data, and the lack of a collaborative tool for analysts and researchers. TimeMerger extends an existing powerful analytical time series tool with a variety of time line visualizations that are suited for many common disparate data types. They are presented graphically aligned to facilitate comparisons, trend analysis, and error interpretation. There have been tools developed and discussed in research that provide portions of similar analytical time lines of different domain data. TimeMerger is unique in its specific visualizations and its application to complex systems and their interactions, and was developed in close cooperation with domain experts.

**KEYWORDS:** Temporal, TimeLine, Visualization, Collaboration, Context

## 1. INTRODUCTION

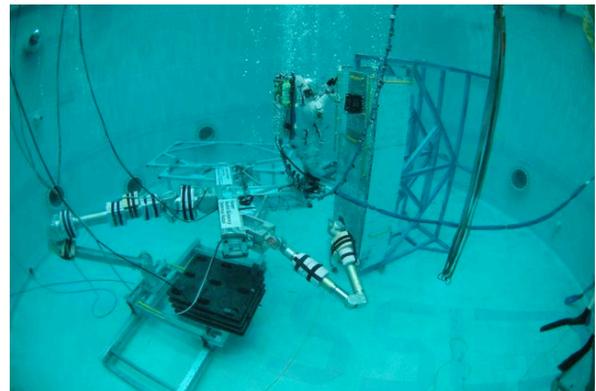
Visualization of temporal data is an important part of the field of information visualization, and has received considerable attention in the literature. Analysis of this kind of data differs from other forms of information visualization in that it does not involve multidimensional analysis measures like clustering, correlation and outliers. Typical tasks addressed include recognizing specific patterns and trends over time, filtering large quantities of temporal variables and detecting anomalies in time series. However, less work has been performed on simultaneously visualizing disparate types of temporal data with the goal of putting the typical temporal queries mentioned above in their appropriate context. This is often required to fully understand the behavior of complex systems which may have states, physical or virtual embodiment, as well as interaction with other systems and its environment. This paper describes TimeMerger, a tool that has different types of visualization interfaces juxtaposed to facilitate understanding of and interaction between variables not previously brought together in a single interface. The primary purpose is to improve analysis of robotic systems

and interactions between such systems and humans by allowing for simultaneous visualization and interaction with the following types of data:

1. continuous: data obtained at regular intervals. In this domain, this is predominantly output from sensors of robotic arm movement.
2. discrete: data that marks events in time. In our scenario, the events are the generation of messages.
3. ephemeral: state values that change and have duration.
4. image: photographs whose source may be video.

## 2. BACKGROUND

The Space Systems Laboratory (SSL) at the University of Maryland has been performing research and development of advanced space and underwater robotic systems for more than two decades. Neutral buoyancy operations with cooperative use of multiple robotic systems and a spacesuit analogy for underwater use are routinely performed. Until recently the data gathered from the different systems in use has not been collected and investigated across platforms. Plotting tools like Excel and LabVIEW have been used on individual systems, but does not easily extend to analyzing multiple complex systems operating in the same environment.



**Figure 1- Underwater Human-Robot Operations in the SSL Simulate Micro-Gravity Conditions**

This is a difficult task as the data collected typically is extensive (logging rates of 125 Hz is not uncommon) and typically of a very disparate nature. Examples include continuous data such as oxygen usage and heart rate of

subjects in spacesuit analogs as well as kinematics and performance data from the robots. A large range of warning and error messages are also produced at high rates, which in this paper is treated as discrete events. Events that have duration typically include state variables like controller modes and frames of reference used for hand controls, but will in general be described as ephemeral events here. In addition, each dive typically produces a large amount of video recordings from several camera platforms typically used to teleoperate the robotic systems.

Understanding this data is crucial to be able to gain insight into the human-robot interaction in particular. This is a relatively immature field and current work in the field includes establishing common metrics with which to compare performance. An effective visualization may aid this work, by providing a means with which to assess and measure these metrics, as well as provide surprising insights into how the systems affect each other and interact. However, analysis of single systems can also benefit from getting a better understanding of the broader context, as in cases where the system is sensing and reacting to a challenging environment, like autonomous underwater exploration and sampling [15]. In addition, the ability to log the behavior of systems has been identified as a critical component to facilitate human responsibility for autonomous vehicles [13].

### 3. DISPARATE TEMPORAL DATA ANALYSIS

In general, exploration of time-oriented data requires some features common to other types of exploration and some unique features:

- Extraction of sub-collections [8]: this allows the user to view subsets of data such as certain error types
- Rescaling [10]: changing the scale of the timeline is critical, similar to a zooming capability in a non-temporal interface
- Documentation [8] – necessary to annotate the events and the discoveries found while performing analysis
- Collaboration [8] – allowing disparate data to come together in a single simplified interface facilitates cross-pollination of specific domain knowledge among the group of experts who may have not interfaced previously

Allen et al [5] suggest that the three main purposes of timelines can be summarized as informing the user directly, showing context, and providing contextual links. In TimeMerger we strive to accomplish this by providing both summary and raw data, displaying data in the context of the

system environment, as well as aggregating data where it increases contextual background comprehension.

Timelines have been used or proposed for a multitude of applications. This includes visualization of medical data [2, 8 and 10]. Aberle et al [2] integrated medical images, related computations, and graphic icons representing textual reports on a common timeline to enhance communication among health care providers. They found that they gained an improvement in the consistency of data entry and computations by presenting the data in a timeline because it gave time a visual dimension. Historical events are also commonly displayed via a timeline such as was done by Allen et al [5].

Timelines have been used as a way to enable operators of Unmanned Vehicles (UV) for exploration and military applications to effectively assess and control multiple vehicles simultaneously in real time. Interfaces have been proposed that predict periods of high operator workload and suggest solutions in addition to presenting mission planning information [14].

## 4. INTERFACE DESCRIPTION

TimeMerger is an extension of the TimeSearcher product built at the University of Maryland [16]. The TimeSearcher tool was selected for its highly regarded time series analysis capabilities. The original TimeSearcher portion of TimeMerger is used for analyzing continuous data such as telemetry data for robotic arms. It was extended to provide new visualizations particularly useful to support the other disparate data types.

### 4.1 TimeSearcher Interfaces in TimeMerger

Continuous linear data can be analyzed using the tools provided by the original TimeSearcher application, shown in figure 2 with telemetry data loaded from the Ranger dexterous manipulator. The graph in the upper left hand corner provides the envelope of the data, and allows for manipulation with different data filtering tools. The scrollable line chart window below it provides the plot for each individual variable against time, while the windows on the right allow the user to select a given variable and view a table of the numerical values. It provides capabilities such as time boxing and the ability to select only variables passing through specific values at a given time, as well as variables with a given gradient. These are features that the users of TimeMerger would also benefit from.

The process of bridging the large gap between the format of the raw data from the robotic arm and the data model required by TimeSearcher meant manipulations had to be designed by the authors. In addition, the ability to dynamically scale the data in this interface was implemented. Other than this extensive work (discussed in

the Implementation Notes section), this interface portion of TimeSearcher was not significantly modified by the authors.

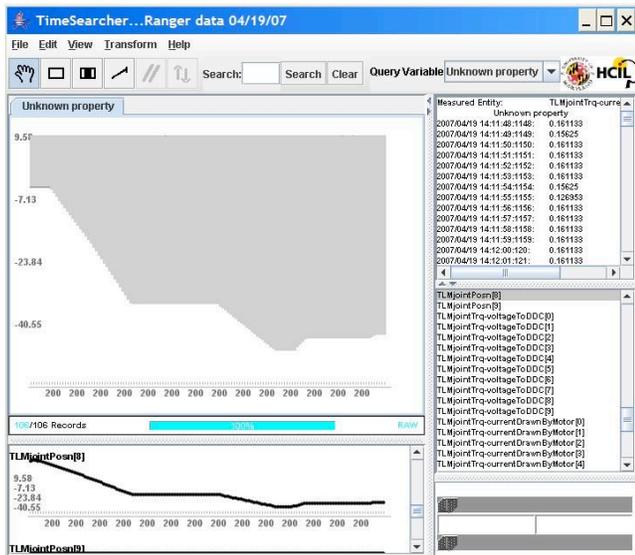


Figure 2 – Main Display Originally from TimeSearcher

#### 4.2 Interfaces Added to TimeSearcher

Besides the required modifications to interface telemetry data to the existing TimeSearcher interfaces, several additional interfaces were developed to extend TimeSearcher to support the other data types necessary with customized displays:

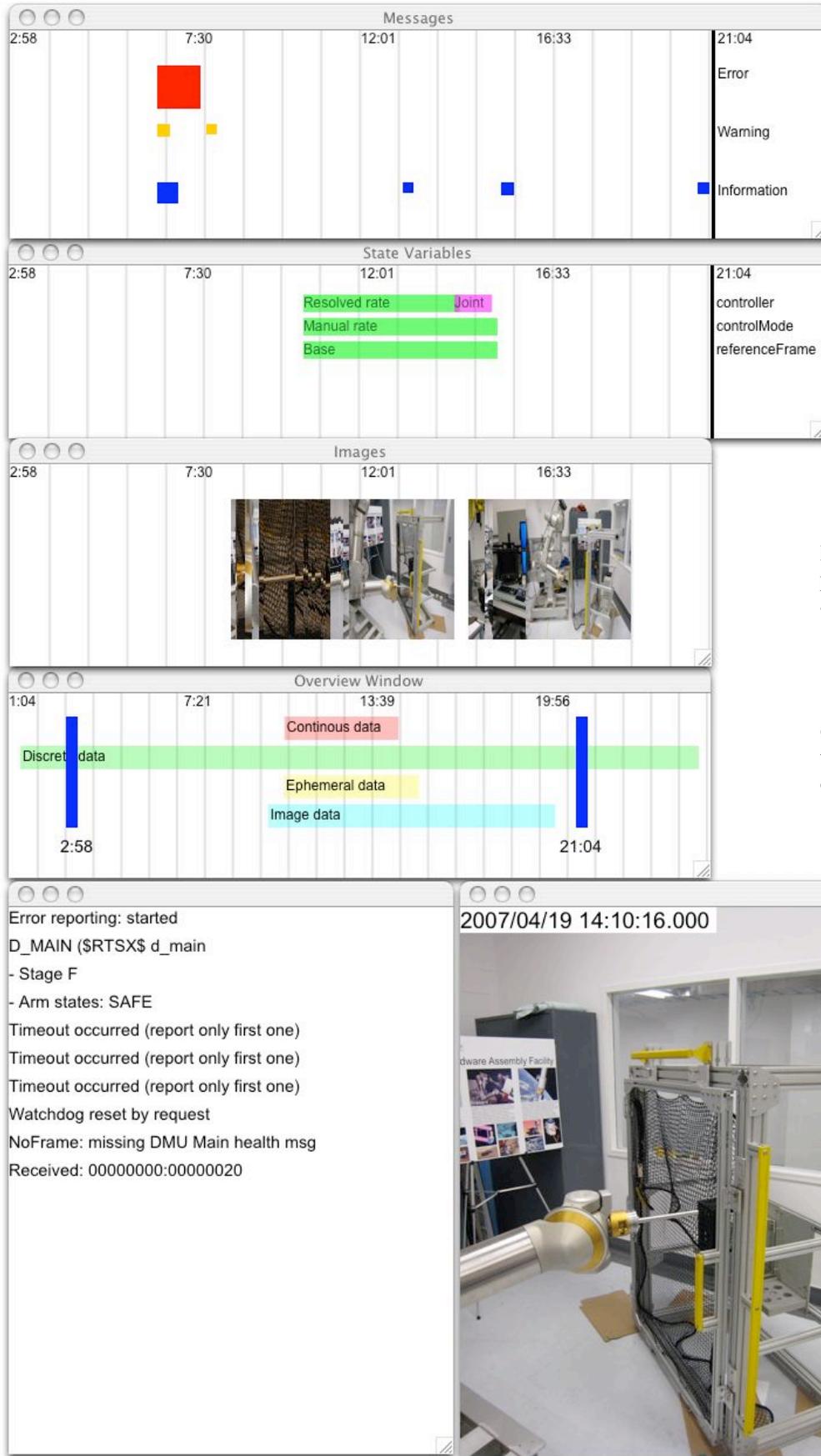
**Discrete Events Display** – this display helps the user determine which time periods to focus analysis on. It readily shows the user the time frames where there were high volumes of error messages (there may be hundreds per second). It contains colored boxes representing aggregations of messages colored by their severity level: red for errors, orange for warnings, and blue for informational (these color associations already existed for the users). The size of the box indicates the number of messages that occurred in the time period, i.e. the messages were binned by their type into regular time intervals. Thus the user can focus on analysis of periods that have large red squares as that would indicate that there were many errors. Selecting a square shows a pop-up text window listing all of the messages that were generated at that time of that severity level. This unique use of the binning concept will aid the user in finding areas of high error rates.

**Ephemeral Events Display** – this display shows the values of selected binary state variables (up to 16 are defined in the Ranger data log) in a Gantt-like bar graph. In other domains the data displayed may be more like typical Gantt chart data plotting the duration of events. Each horizontal row of bars represents the changing state of a variable. Each bar is labeled with the value of the variable. Bar coloration changes as a state variable value changes to reduce search required to find values. The variable name appears on the right of the display to facilitate alignment of the windows along a common timeline.

**Images Display** – this display shows photographs at various intervals during the run. The images are thumbnails that can be selected for larger viewing [2]. Shown in Figure 3 is a picture of the Ranger robotic manipulator using one of the interchangeable end effectors (in this case a wrench type effector) to open a door by unlatching a J-hook teleoperated by our main adviser. Photos were taken during a run that the authors participated in at the Space Systems Laboratory at the University of Maryland.

**Overview Display** – The Overview slider gives the user a manipulable medium to explore subsets of all data available. All displays contain the subset of data from the same period of the epoch. The epoch begins at the earliest time that is found across all data files read in and ends on the latest such time. The overview display shows the user where the current period is in relation to the epoch. This interface also shows the user the periods of data available for each of the different types of data as all data files may not span the complete epoch. To change the current period, the user simply slides the beginning and ending slider bars to the desired time. After the user moves the sliders, all windows are rescaled immediately. This interface is a novel design that provides a critical requirement of information visualization tools: overview. This then allows the user to drill down to get details on demand for different types of data as necessary.

In all windows, gray vertical lines (barely visible in the screen shot) indicating 1-minute intervals help the user's time orientation: as the time period becomes smaller, the number of gray lines decreases and the width between them increases. This provides a sense of scale to the displays [9].



**Discrete Display** shows aggregation of messages by their type, sized according to volume

**Ephemeral Display** shows values of state variables

**Images Display** shows pictures taken throughout a run

**Overview Display** allows the user to change the time period displayed in all other windows

**Message Details** Text of messages are displayed when the user selects a square from the Discrete Display

**Enlarged Image** is displayed when the user selects a thumbnail image

Figure 3 – Added TimeMerger Displays

## 5. USERS AND TASKS

The projected users of TimeMerger are researchers, engineers and students of the SSL. The tool was also aimed at users with less detailed knowledge about the specifics of the systems, for example graduate students doing research in Human-Robot Interaction (HRI) and related fields. By providing a tool that can help give context around a specific set of data of interest, it was hoped that these students could explore and learn about the interaction between systems and between the state of the system's internal workings and its environment. The addition of an annotation feature would further expand the use of the system to allow experienced and knowledgeable users to pass on beliefs of important events and causalities, and store this information with the actual data.

Research in the SSL typically involves a multitude of robots, including free flying camera platforms, dexterous manipulators of varying forms and capabilities as well as surface rovers. Other work focuses on spacesuit analogies and human performance, adding robotic capabilities to humans (exoskeletons for rehabilitation, spacesuit adaptation and grasp assistance) as well as quantifying and improving input devices for teleoperation. For sufficiently complex systems like a single Ranger manipulator, the data generated is in itself suitable for analysis with TimeMerger, including tasks like:

- Review commonalities of events that occur preceding a reoccurring error for possible patterns.
- Analyze and debug erroneous behavior, for example investigate changes in actual commands sent to a robot when switching input control frames of reference
- Assess causes for autonomous decisions made by the robot, for example decisions to “safe” the arm and ignore further user control input

However, a lot of the work being performed in the SSL involves multiple systems acting together. For example, humans and robots working collaboratively on simulating tasks such as servicing the Hubble space telescope. Here tasks suitable for TimeMerger include:

- Assess and compare effect of scheduling (detailed timeline of tasks performed by robots and humans) on human workload (heart rate, heat rejected) and critical incidents (near-collisions, communication errors, robot motion singularities)
- Post-dive analysis and review, stored for comparison across dives and educational purposes
- Get insights into causalities across systems, for example how performance of teleoperator of a given robot is affected by camera views provided by another system

## 6. SCENARIO

As part of the development of TimeMerger, the authors participated in operating the Ranger robotic manipulator in a 1-G environment, meaning that it was not operating in the neutral buoyancy tank where gravity is compensated via flotation devices. This allowed for observation of the process of running the system, analyzing the current software tools utilized and generating a real and complete set of test data. In this run, the robotic arm was equipped with a bolt-drive tool and the arm was used to open a door latched by a J-hook, similar to what is found on the Hubble space telescope.

The data collected involved all four types of data that TimeMerger can visualize. The full run took approximately 26 minutes, with errors and warnings being logged at all times, as shown by the green labeled bar in the overview window. Most of these system messages were generated early in the run, when the operator was adjusting the manipulator's settings to prepare for the task. When this was completed, the operator started logging data for continuous and ephemeral data, as can be seen in the same window. Images were also now taken, and continued for some time after the main task had been completed.

Thus the overview window gives the analyzer an idea of what time periods contain data of the various types, which can help guide exploration and analysis. Looking in more detail at the continuous data collected, the data envelope immediately reveals plateaued changes in some variables which may be of interest. Using the timebox feature, the variable can be isolated as `TLMjointPos[8]`, which is the telemetry reading the joint angle of the 9<sup>th</sup> joint on Ranger, here the tool drive. Using the sliders to focus on an area of large changes, the images now displayed for this period and the lack of warning messages confirm the fact that the tool drive is actually attempting to manipulate one of the J-hooks at this time.

Further exploration reveals that the voltage (variable `TLMjointTrq-voltageToDDC[8]`) applied to the motor is not constant, suggesting either undesired internal friction in the drive, or complexities in the interaction with the J-hook. After limiting the scope to just the few seconds the drive is manipulating the J-hook, the images reveal that the drive does not immediately connect with the hook, and that abrupt changes in voltage occur at the moment of capture and maximum rotation of the hook.

## 7. DESIGN PROCESS

Our main adviser was a member of the SSL lab team who has the broadest knowledge of the details of the domain, the processes, and the data that pertain to the runs executed and analyzed in the SSL. He is the lead engineer responsible for the advanced Ranger robotic system and the software architecture used in most systems in the laboratory. He was involved from the beginning and

throughout the design process for TimeMerger as suggested by Noyes et al [7]. Both direct methods, such as interviews, and indirect methods, such as observation, were used to elicit knowledge about the domain and the specific details of the problem [7].

## 7.1 Initial Design Meeting

The initial hour-long meeting provided a basic understanding of the domain, the current problems, and some of the desirable features in a system that would facilitate analysis of temporal data from disparate sources. The task of analyzing the output of a run is made difficult by several factors:

- There are many different sources of data and each of the sources represents a different kind of data that is best visualized in a specific way.
- Runs are typically short in duration, so it is difficult to analyze during execution.
- Large volumes of data are generated that is hard to assimilate.
- There is currently no method to compare the disparate data on a single timeline together to see the correlation of events.

The authors learned in this initial meeting that there are many different kinds and sources of data that need to be analyzed. A focus would be on telemetry data from the Ranger robotic manipulator, as it was the most readily available and provided a suitable visualization challenge with its logging of 127 variables at a rate of 125 Hz. The authors realized at this time that the analysis of this voluminous data would best be served by the capabilities already available in the TimeSearcher tool.

The other types of data generated from a run would require customized displays. Many of these data sets were presumed at this point to be best displayed via a Gantt chart-like type of display, but this evolved throughout the design process.

The two primary categories of this data were thought at this point to be discrete events and ephemeral events and were so labeled. For the initial design, the discrete events would be displayed as markers on the timeline. An example of a discrete event might be “Diver enters the tank”. The ephemeral events would be displayed in a Gantt chart. Ephemeral data was so named because the events had short durations. An example of such an event would be if the robotic arm “safes” (i.e. ignores further control input from the user and holds its position) for 5 seconds. This would produce a finite length event until the operator commands the robot to resume its normal operation.

In addition, it was considered desirable to include images of events to document visually what is happening during a run. The addition of this aspect makes interpretation of

data significantly easier as it serves as a pictorial documentary of what happened at what point on the timeline. Ringel et al [16] proved the value of landmarks on a timeline to improve search capabilities due to the fact that human memory tends to be organized in episodes. For historical documentation purposes and for long runs, the images could aid the users in such a similar manner.

## 7.2 Presentation of Initial Prototype

After this first meeting, the authors implemented a prototype of the interfaces described above. In a subsequent meeting, this same domain expert spent several hours reviewing this initial prototype, providing specific feedback, and even allowing the authors to help with a test run to generate real data. Feedback effected several displays but most prominently the discrete event display dramatically changed. He cited the following additional capabilities as being desirable:

1. In the discrete events display in this first version of TimeMerger, the point at which events occurred was marked by a colored icon and labeled accordingly, for example “robot arm loses power”. In this meeting, the authors learned that these events were really different levels of error messages that came from one data file. It was also learned that these can be generated at such high volumes that the current interface would not be able to accommodate all such messages as it was designed. Our adviser also explained that it is not desirable to see the details of all messages. He did state though that it would be useful to use this information as a means of determining the periods of time that most require analysis, and suggested that it would be beneficial to have the capability to see at a glance the periods in which high volumes of warning and error messages occurred. This overview capability would serve to focus the user’s attention to attempt to determine the source of the problem in that time period first. It was learned that there are 3 distinct types of messages: error, warning, and information. Instead of the original Gantt-like display, it was suggested that messages be grouped into time periods by their type.
2. Also in the discrete events display, the ability to search to find all errors of a particular type as specified by a substring in the message text was requested. The authors envisioned a text box interface for the user to specify the string desired. After entering the text, any time periods that contained an error of that type would be marked in some way on top of the colored square in black. This feature was not implemented due to time constraints.

3. The domain expert also suggested that it would be useful to select one of the boxes to see a text window that contained the exact error messages that occurred within that time period. This details on demand capability was provided before the final prototype review.
4. A short term memory problem for users could be solved by translation of error codes to the corresponding text. Contained in the messages are numeric codes that have a correlating body of meaningful text. While this data was obtained from the user after this meeting, it was not incorporated before the next prototype review.
5. The user suggested that viewing the images in the timeline as thumbnails would be sufficient if the capability to enlarge them were available (as also suggested by [2]). Therefore, before the next review, the photographs were scaled down in size and the capability to click on an individual picture and subsequently view the image enlarged was added (another example of details on demand).
6. Another request was to have the ability to select the list of variables displayed in the “Ephemeral” window. It was learned during this review that there are exactly 16 different state variables. The user suggested that with different types of analysis, different variables are of interest, but never all at once. A pull-down menu of the complete list of variables could be provided to allow the user to select the subset desired. This capability was not addressed.
7. A “sticky note” feature that the authors suggested as possibly useful was regarded by the domain expert as not being a high priority. The authors thought that it would be very desirable to annotate the events of the run along the timeline at a high level for various sized time periods that represent a logical unit. It was envisioned that this would store the user’s interpretations and observations, for example “peak in current drawn by manipulator is due to unbalanced arm”. This feature was not given a high priority but is still considered desirable.

### 7.3 Final Prototype Review

In the final prototype review, three advisers viewed a demo of the TimeMerger application on a large four-paneled display (each 1680 by 1050 pixels) in the SSL laboratory. Our domain expert was joined by two potential users, graduate students with interests in using TimeMerger. This new version used data generated from the run conducted during the previous meeting, included the added capability to rescale all displays through a single bar slider type of interface and showcased modifications to the new displays.

The new slider interface, the “Overview Display”, allowed a user to define the period of data of interest consistent across all displays. This novel feature was found to be useful by all and key to zooming in (more appropriately called “rescaling” in temporal applications [10]) to the period of interest necessary for analysis.

Also in this version, the discrete events display was modified as requested by the main adviser to show the messages binned by type in time periods. The squares are colored according to the users’ existing color association as much as possible. In the system that is used currently to view the messages as they occur, warnings are colored yellow. Yellow was deemed a difficult color for visibility, so the close alternative of orange was used instead. The user did not have a suggestion as to how the time period for the bins was determined, as there is no meaningful time period. This was arbitrarily determined based on the length of the time period. This concept of binning the messages we feel will be very useful to the users in focusing their attention to the most critical time frames and is a novel utilization of this visualization technique.

During the end of this development phase, it had been determined that the events displayed in the ephemeral window were in actuality simply state variables. This meant that the Gantt chart interface was somewhat over-designed for this domain as it allows for staggered values displayed on sequential horizontal lines. This was discussed with the advisers and they agreed that these variables are indeed binary state variables. This interface had also been retained as originally designed however due to an occlusion problem. Specifically, the state variable changes occur at such short time intervals that displaying a colored bar for each state value on a separate line with a label had the benefit of solving the problem of labels being overwritten on top of each other. In theory, the mutually exclusive values of these state variables could simply be placed as dots on a time line or a single bar that changes color if the labels were laid out in some other fashion such as alternating above and below the line or bar.

The “sticky notes” idea was discussed and the group decided that it would be useful as a separate display that contained annotations that would line up along the time line. They advised that it would be preferable to have a single such window that collected all notes rather than allow notes along the timeline in the various displays. They also decided that annotation of the photographs was not necessary (as was currently implemented) because this would not be available as data and the sticky notes could be used for a similar purpose.

It was suggested by one of our advisers that perhaps a user could click on a point on any time line and see a list of all the events that are happening at that time.

Another useful comment provided was that the labels that appear in some of the displays above a line of information

could be placed along the side outside of the area where the data was displayed. This idea would greatly improve the vertical alignment of the displays such that they would be more compact and therefore easier to compare (this was implemented after this discussion and is shown in figure 3).

An especially novel idea came from one member of the advising group who suggested that it might be useful to compare the events of two time periods. This idea would add a new dimension to the analytical capabilities of the application that had great potential benefits. However, doubling the screen real estate requirements would only be viable in such circumstances as the demo took place: using four oversized monitors. Indeed, the authors' laptop screens were already almost completely utilized. It was agreed that this feature should be considered as it could be beneficial in comparing similarities not only from one run to another of the same type, but also of different types to determine similarities between neutral buoyancy and 1G runs for example.

The primary supervisor of this project suggested that after addressing these smaller types of tweaks to the software, that a more complicated additional component that would be desirable would be a view of the robot in a 3-D simulation software package available in the SSL, using the actual kinematic data recorded (angles of each joint). This would provide a good indication of the state of the robot itself at all times with respect to a reference frame, but would have to be supplemented with images to provide information on its surroundings.

#### **7.4 Continuing Development**

While this design review was considered to be possibly the last, development on the prototype continued. The emphasis was placed on refining the details of each display. Particularly, the overview slider used to select the time period was refined to include colored bars that indicate the time period of data that is currently loaded for each of the different data types. This is a characteristic of this domain that may not be common with all temporal data: data may not be available for all time periods within a complete run. This may be due to sensors losing power or not being utilized during the entire duration of a run, as well as systems being used in a dive at different times, often limited by human fatigue or battery limitations. This design feature is a unique visualization that we hope will greatly aid the users in their selection of time periods for analysis.

As a related aside, a design issue that was monitored along the course of the development cycle was that of the possibility of the user having multiple files of the same data type that may need to be loaded simultaneously. This issue has not been entirely resolved. It would seem that there might be different cameras taking pictures at different locations around a run. This would require the capability to have N number of images windows, each containing a different set of pictures. However, it was assumed with

reasonable certainty that the error messages and the state variables would all be contained in a single data file for each type and that it would not be necessary to have multiple instances of those interfaces.

The overall design review process was successful for many reasons. The advantage afforded by having an enthusiastic, knowledgeable, and articulate adviser who was generous in giving time to this work was an immeasurable benefit that was key to this project's success. The availability of other domain experts helped to provide a balance of different perspectives and unique ideas that is necessary for the successful development of any software application.

#### **8. IMPLEMENTATION NOTES**

TimeSearcher 1.3 was selected as opposed to newer versions of TimeSearcher because it was written in Java and included the angular search feature described above even though this older version did not have the scaling capability. Therefore, this function needed to be added in. Although this process at first seemed simple, the actual implementation proved difficult as there was little documentation and no defined API. In the end this was solved by effectively limiting the subset of data visible to TimeSearcher, and the feature is now close to full functionality. As TimeSearcher was not made with dynamic scaling in mind, there are still some issues with updating data axes and displaying normalized data. The work performed trying to mend these issues was considerable, and due to time constraints the authors had to abandon resolving all of them.

Implementation of TimeMerger started very slowly. The authors had many technologies to learn: Java, piccolo [19] and Swing (for graphics), eclipse (Java building environment), and TimeSearcher. Starting with a base of code is a double-edged sword: it provides an existing framework of code and utilities that provide a solid basis, however, the task of learning the complex TimeSearcher code was difficult. Attempts to incorporate the TimeMerger displays seamlessly into the existing window hierarchy were not fruitful and were abandoned after much analysis (and subsequent advice from one of TimeSearcher's authors, Harry Hocheiser). Piccolo is a structured graphics framework [19]. It is a toolkit that allows you to quickly build graphical user interfaces that are efficiently rendered. It was straightforward to use with sufficient examples and a consistent API. One bug that was discovered but quickly fixed by Professor Ben Bederson of the University of Maryland was in the PFrame implementation used for creating our displays. This bug was causing a portion of the window to not be filled with its graphic objects until the window was resized manually by the user.

Data preparations were difficult to do and some compromises for the purposes of demonstration needed to

be made. The robot arm telemetry is sampled at a rate of 125 hertz. During the sample scenario that we ran, 10 minutes of sample data were generated, creating a 50 Mb data file. We wanted to analyze the whole sequence of data, therefore data was sampled from data files via a script that extracted every 125th record so that there would be one record per second. To be utilized in TimeSearcher, this data also had to be transposed and manipulated to meet the other requirements of the TimeSearcher file format. Shell scripts were written to automate this process to increase the turnaround time for future analysis.

## 9. BENEFITS OF TIMEMERGER

Applying the six basic ways that information visualization amplifies human cognition [12], it is apparent that TimeMerger fulfills each in some way:

| <b>How Information Visualization Amplifies Cognition [12]</b> | <b>TimeMerger Example (not including benefits provided by TimeSearcher)</b>  |
|---|--|
| <i>Increase cognitive resources</i>                           | Provides visual data aggregations to increase working memory   |
| <i>Reduce search</i>  | Represents large amounts of data in a small space that would have required search in different locations                   |
| <i>Enhance recognition of patterns</i>                        | Vertical alignment of data at the same time point shows time relationships   |
| <i>Support easy perceptual inference of relationships</i>     | Side-by-side comparison of different attributes aides correlation of variables/states                                      |
| <i>Perceptually monitor large quantities of events</i>        | Large volumes from different sources are aggregated compactly enough to increase amount of events simultaneously monitored |
| <i>Provide a manipulable medium for exploration</i>           | Overview slider provides control of time period analyzed   |

**Figure 4 - Desired Effect of TimeMerger on Cognition**

Many benefits of TimeMerger are suggested by the authors, the consultants on this project, and research in the area of timeline visualization:

- Facilitate discovery of relationships [8], anomalies and trends [9]
- Reduce likelihood of information being overlooked [9]
- Visualize information too voluminous to be analyzed in raw format

- Facilitate access to detailed information
- Provide customized visualizations that amplify meaning
- Enhance collaboration among domain experts of different areas of data

## 10. FUTURE ENHANCEMENTS

In addition to enhancements previously mentioned throughout this paper, the following potential future enhancements that come from both the potential users of this system, the authors, and from literature research of applications with similar requirements are suggested:

- Add the capability to receive real-time data
- Play back capability with a single-stepping debugging feature[4]
- Summary information [2] for a run (e.g.: total number of messages of each type, elapsed time, high-level summary text file, run name)
- Icons that represent hypertext-style links to raw data
- Ability to view images as a movie[2]
- Ability to display images spaced in an equidistant layout rather than according to the time interval for improved viewing [5]
- Increase capability for TimeMerger to be used by other domains, for example medical

## 11. CONCLUSION

A primary goal of TimeMerger is to provide context for disparate data types. It accomplishes this by associating all data with a common timeline. As an information visualization tool, TimeMerger provides overview, details on demand, and zooming (rescaling). Particularly the Overview Display and the Discrete Display are novel in their design for this domain. The former provides a manipulable medium for exploration of the disparate temporal data types and their position in the epoch. Messages are binned in the Discrete Display to minimize cognitive load and focus the user on periods of interest. The tool is aimed at analyzing voluminous amounts of data from complex systems and aiding collaboration among researchers and analysts.

Feedback from experts both inside and outside of the SSL have been encouraging: not just because of these features but because of the void of such capabilities in this domain. The authors feel that this project proved that a tool can be built that will fully accomplish the original goals and that a meaningful subset of those requirements have been successfully implemented in TimeMerger.

## 12. ACKNOWLEDGEMENTS

We would like to thank Dr. Ben Shneiderman for his guidance. Also, Professor Ben Bederson deserves a special thank you for his quick fix to piccolo. Harry Hocheiser thankfully provided the TimeSearcher code to us which gave a solid foundation for this application. We appreciate his work on TimeSearcher as well as the other developers before and after that have worked on it. The authors would especially like to thank the users that reviewed TimeMerger: Stephen Roderick (lead software/robotics engineer), Shane Jacobs (EVA team-lead), and Sharon Singer. Special appreciation goes to Stephen for his enthusiastic participation, insightful ideas, unending help, and infinite domain knowledge. It wouldn't have been possible without him.

## REFERENCES

- [1] Monroy, C., et al., Interactive Timeline Viewer (ItLv): A Tool to Visualize Variants Among Documents. LNCS 2539, p. 39-49, 2002. Springer-Verlag Berlin Heidelberg, 2002.
- [2] Aberle, D.R. et al., Integrated multimedia timeline of medical images and data for thoracic oncology patients. Radiographics, vol. 16, number 3, p. 669-681, 1996.
- [3] Monroy, C., Furuta, R. and Mallen, E., Visualizing and Exploring Picasso's World. Proceedings of the JCDL 2003 Conference, Houston, Texas, 2003.
- [4] Szekely, P., Rogers, C.M. and Frank, M. Interfaces for Understanding Multi-Agent Behavior. Proceedings of the 6th international conference on intelligent user interfaces, Santa Fe, New Mexico, pp. 161-166, 2001.
- [5] Allen, R.B, A Focus-Context Browser for Multiple Timelines. Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries, Denver, pp. 260-261, 2005.
- [6] Ogden, P.G., Human computer interaction in complex process control: developing structured mental models that allow operators to perform effectively. People in Control - Human Factors in Control Room Design, pp. 120 -125, Manchester, UK, June 2001.
- [7] Noyes, A.M., Star, A.F., Working with users in system development: some methodological considerations. IEE Colloquium on 'Integrating HCI in the Lifecycle', pp. 7, London, UK, April 1995.
- [8] Goren-Bar, G., Shahar, Y., Galperin-Aizenberg, M., Boaz, D., Tahan, G., KNAVE II: The Definition and Implementation of an Intelligent Tool for Exploration and Visualization of Time-Oriented Clinical Data. Proceedings of the working conference on Advanced visual interfaces, pp. 171-174, Gallipoli, Italy, 2004
- [9] Shabtai, A., Klimov, D., Shahar, Y., Elovici, Y., An Intelligent Interactive Tool for Exploration and Visualization of Time-Oriented Security Data, Proceedings of the 3rd international workshop on Visualization for Computer Security VizSec '06, ACM Press, November 2006.
- [10] Plaisant, C. Milash, B., Rose, A., Widoff, S., Shneiderman, B. LifeLines: Visualizing Personal Histories. CHI April 13-18, 1996
- [11] Thomas, James J., Cook, Kristin A., editors. Illuminating the Path: The Research and Development Agenda for Visual Analytics. US DOH National Visualizatin and Analytics Center, 2005.
- [12] Card, S.K., MacKinlay, J.D., Shneiderman, B., Readings in Information Visualization: Using Vision to Think. Morgan Kaufmann, San Francisco, CA, 1999.
- [13] Shneiderman, B., Human Responsibility for Autonomous Agents. IEEE Intelligent Systems: Interacting with Autonomy, Vol. 22, Nu. 2, p. 60-61, 2007.
- [14] Cummings, M.L., Brzezinski, A.S. and Lee, J.D., Operator Performance and Intelligent Aiding in Unmanned Aerial Vehicle Scheduling. IEEE Intelligent Systems: Interacting with Autonomy, Vol. 22, Nu. 2, p. 52-59, 2007.
- [15] ASTEP [Internet], Dexterous Robotics at the Space Systems Laboratory. <http://robotics.ssl.umd.edu/astep/index.shtml>, accessed 04/31/07.
- [16] Visual Exploration of Time Series Data [Internet], Human-Computer Interaction Lab, University of Maryland. <http://www.cs.umd.edu/hcil/timesearcher/>, accessed 04/31/06.
- [17] Ringel, M., Cutrell, E., Dumais, S., Horvitz, E., Milestones in Time: The Value of Landmarks in Retrieving Information from Personal Stores. INTERACT 2003, September, 2003.
- [18] Bayer, S., Damianos, L., Kozierok, R., Mokwa, J., The MITRE Multi-Modal Logger: Its Use in Evaluation of Collaborative Systems. ACM Computing Surveys, Volume 31, second issue, 1999
- [19] Piccolo: A Structured 2D Graphics Framework [Internet], Human-Computer Interaction Lab, University of Maryland. <http://www.cs.umd.edu/hcil/piccolo>, accessed 05/07/07.