Proceedings of the 1995 Winter Simulation Conference
ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman

THE POWER AND PERFORMANCE OF PROOF ANIMATION

Nancy J. Earle
James O. Henriksen

Wolverine Software Corporation
7617 Little River Turnpike
Annandale, Virginia 22003, U.S.A.

ABSTRACT

The Proof Animation™ 2.1 product family provides our customers with the tools for animating a vast array of applications. There is a version of Proof Animation to meet any need, from small and mid-sized projects to large scale applications. This product family runs on readily available, inexpensive PC hardware without the need for special hardware. Proof Animation is ASCII file-driven and features a general purpose design. Its vector-based geometry provides a large animation canvas with the ability to zoom in or out while maintaining crisp, clear images. Proof Animation's features include post-processing for maximum performance, built-in drawing tools and CAD import/export for ease of creating animation layouts, dynamic bar graphs and plots used for displaying statistics, a multi-windowing display, and a unique presentation-making capability. Because of its open architecture, Proof Animation can serve as the animation tool for models written in a wide variety of simulation and programming languages while remaining easy to use. Proof Animation's user interface is menu-based and can be accessed using either a mouse or keyboard. Its superior power and performance assures smooth, realistic motion for animations regardless of the size, complexity, or application.

1 INTRODUCTION

Animation software has been around for many years; the concept is not new. Animation is often considered a requirement to a simulation study because it is a meaningful, proven way to show results to an audience with varied backgrounds. As more animation products enter the market, a user must make sure that the product purchased can meet the task at hand, as well as any future projects. Some animation software is limited to a single type of application. Moreover, these animation packages may only be useful for a specific type of problem *within* that application. In almost any application, there are many areas that benefit from simulation and animation. These can range from capacity analysis to scheduling, staffing, and more. Some application-specific animation software may not be able to handle the full range of problems even within a *specific* application. Buying single-application animation software can be costly, especially if a user does projects within a wide range of applications.

Proof Animation is a powerful general purpose animation tool. It is not tied to a specific area or application. Proof Animation can be used to animate the full range of applications from areas such as Business Process Reengineering to the classic applications such as healthcare, manufacturing, and traffic. The size of the system to be animated is not an issue when using Proof Animation. There are versions of Proof Animation that can be used to animate small to mid-size systems as well as a version to handle the largest systems.

A user may want to add animation to an already existing model. Since time is rarely unconstrained, he or she does not want to have to change any existing and valid model logic to make the animation look realistic. Some animation tools are so limited that it is difficult or even impossible to produce an animation that can mimic the motion of the real-world system. Wolverine Software Corporation's goal has always been to design animation software that is easy to use and able to produce realistic animations that help sell the results of simulation projects.

Another issue in animating existing models is interoperability. Can the animation software work with the existing simulation software? Proof Animation can be paired with most simulation and programming software. The user can continue to use the modeling tool with which the existing models were written.

Proof Animation 2.1 has an advanced feature set unparalleled in its ability to meet the challenges of animating today's complex systems while keeping ease of use a top priority. In the following sections we describe the products, discuss their design, give an

overview of how they are used, and describe the advanced features.

2 THE PROOF ANIMATION FAMILY

All of the Proof Animation products run on readily available and inexpensive PC hardware. All run as 32-bit applications that require only a 386 or better CPU, a math coprocessor, and at least a VGA-compatible video card. Because this basic configuration is very widely available, the portability of the animations is maximized. It is very likely that this hardware configuration will be available at any site.

The Proof Animation family runs under DOS and can also be launched as a full-screen application under Windows 3.1 and higher or OS/2 2.0 and above. A set of Windows icons are supplied with each of the Proof Animation products. The following products comprise the Proof Animation family:

- PROOF ANIMATION
  Proof Animation is the basic animator. Memory size is fixed and limited. Includes built-in CAD import/export feature. Good for small to mid-sized animations.

- PROOF PROFESSIONAL
  Proof Professional exploits all available extended memory for animating large systems. Includes built-in CAD import/export feature. Included at no additional cost are our 1024 X 768 High-Resolution version and our High-Color Proof Animation that runs in 640 X 480 resolution. Both the High-Color and High-Resolution versions require a super VGA video card.

- RUN-TIME PROOF PROFESSIONAL
  Run-time Proof Professional runs developed animations or presentations, but has no animation *development* capabilities. It provides a low cost way to run different scenarios with a fixed layout file prepared using Proof Professional or Proof Animation.

- STUDENT PROOF ANIMATION
  The student version of Proof Animation is included with the *Using Proof Animation* text. Size and playing time limitations are imposed; otherwise it is identical to Proof Animation.

- PROOF ANIMATION DEMO MAKER
  Demo versions of animations can be prepared under a licensed copy of Proof Animation or Proof Professional containing the Demo-Maker add-on. Copies of the executable demo files can be reproduced and distributed free of charge and viewed by anyone. No licensed copy of Proof is needed to *view* a demo prepared with the Demo Maker.

3 MAXIMUM PERFORMANCE DESIGN

3.1 The General-Purpose Approach

One of the ways in which the performance of Proof Animation is maximized is its general-purpose design. A general-purpose animation product can be used without being tied to a specific simulation or programming language. While built to work easily with Wolverine's GPSS/H simulation software, Proof Animation also provides affordable and powerful animation software to users who develop models in other simulation and programming languages.

Most animation software from other vendors is directly coupled to their simulation software. In other words, one cannot use *their* animation software without also using *their* simulation software. In some cases, the simulation and animation software are sold only as a pair, so both must be purchased regardless of the needs of the user. The suggested advantage of the coupled approach is that because the animator has direct access to the simulation events, development of the animation is supposedly simplified. However, the real advantage is felt by the vendors. They sell more software since their users do not have an option to pick and choose based on their needs. Moreover, a user of the coupled software has little or no control over what information is passed to the animation; therefore, he or she may actually have to alter the modeling approach used in the simulation to achieve the desired appearance and level of detail for the animation.

Another disadvantage of the tightly coupled simulation/animation package is cost. Sole sources tend to be expensive. Vendors of these tightly coupled packages often claim that their approach is the *only* way to add animation to a simulation. Proof Animation has proved that wrong. The number of success stories using Proof Animation with other software continues to grow. Furthermore, a benefit of the mix-and-match strategy for software purchases is that the selection can be based on optimal functionality and price.

3.1.1 ASCII Input Files

Proof Animation can be used with other software because of two major design features. One is that Proof Animation is driven by ASCII files. Therefore, any

software capable of writing ASCII text files can be used with Proof Animation.

Proof Animation requires two ASCII input files to run an animation: the layout file and the trace file. The layout file describes the geometric details of the background over which objects move, provides geometric definitions and properties for such objects, and defines logical paths along which the objects move.

Ordinarily, layout files are produced at least in part by using Proof Animation's drawing tools; however, the layout file command set specifications are published so programs can easily be written to generate layout files. For example, some users have written front ends for their simulation models that allow different system design parameters to be specified for each run. Based on these parameters, different geometric configurations are written and incorporated into a layout file. The new layout appears on screen when Proof Animation is invoked.

The trace file contains a time-ordered sequence of commands such as CREATE, DESTROY, PLACE, PLOT, MOVE, SET SPEED, SET COLOR and many more. This file provides Proof Animation with information on when, where, and what to create, destroy, place, plot, etc. Trace files are free-format, and the commands are easily learned and used. They provide exactly the kind of flexibility necessary to easily be integrated with the simulation model logic. Any language that can produce formatted ASCII output can write a trace file.

3.1.2 Post-Processor Animation

The second of the two design features that make Proof Animation compatible with other software is that it is a post-processing animation package. Post-processing means that it runs *after* the simulation has executed. Both the layout and trace files must exist before invoking Proof Animation. They cannot be written and read concurrently.

Two great advantages result from the post-processing approach. First, PC hardware is not shared between the simulation and the animation. This leaves the entire CPU for running the animation. Second, it provides the abilities to jump back and forth in time during the animation playback, to speed up or slow down the viewing speed, or show all or a specific portion of an animation. These features make it easy to investigate unusual system behavior or highlight points of interest.

3.2 Vector-Based Geometry

In the Proof Animation product family all layout file information is based on vector geometry. Vector-based descriptions are automatically mapped into the screen's pixels to build the image. One of the advantages of this approach is that layouts can be much larger than a single screen. With the ability to zoom in or out and pan, larger layouts are easily navigated to show the overall layout or zoomed in to whatever level of detail is necessary. Vector-based geometry also provides the ability to have moving objects realistically *rotate* around corners instead of the sliding effect to which other animation packages are limited.

Another advantage of vector-based geometry is that many CAD packages are capable of producing standard vector-based .DXF files. In many cases, a CAD drawing already exists for the system to be animated. If that is the case, the effort of redrawing an entire layout can be avoided. Proof Animation's built-in CAD Import/Export feature provides the capability to convert industry-standard .DXF files into Proof Animation layout files, *and vice versa*. Credibility of the study is enhanced when viewers see a familiar CAD drawing of the system integrated into the animation. These advantages maximize the power of the animation by giving a user total flexibility on the detail and complexity of the drawing.

Instead of vector geometry, animation packages may use a pixel-oriented approach for drawing. With the ability to manipulate individual pixels, one can produce detailed images. However, this level of detail is time consuming to draw and can often be lost because of the scale at which the animations are viewed. Some other disadvantages of pixel-orientation are: (1) pixel-oriented images cannot be rotated; (2) layouts are often confined to single-screen images. Some animators offer multi-screen operation; however, the individual screens are disjoint and independent, unlike Proof Animation's single, continuous canvas; (3) zooming in on pixel images magnifies the jagged edges inherent in all such images. When a zoom in is performed in Proof Animation, the vector-based image maintains its crisp and clear appearance. Lines continue to look like lines. If a zoom in is performed on a pixel-based animation layout, the effect of the jagged edged image makes a line look more like a stairway.

3.3 Smooth Motion

The maximum performance design of Proof Animation achieves very smooth motion. Proof Animation maintains this smooth motion by updating or refreshing the screen 60-70 times per second. Other software can often sustain refresh rates of only 5-10 updates per second. The ultimate purpose of an animation is to achieve a realistic depiction of the system being studied, allowing the audience to gain confidence in the results of the simulation study. Objects that move smoothly

across the screen are more realistic than those that jump across the screen.

4 NAVIGATING THE MODES AND MENUS

Proof Animation is organized into seven menu-driven *modes*. Each mode is a collection of closely related functions. Switching between these functions is very easy. Usually, a single mouse click is all that is required. Switching between modes is also easily done, but it implies major changes of context. For example, running an animation and drawing a layout are vastly different activities. Each mode has one or two main menu bars at the top of the screen. Clicking on main menu items invokes the options for the lower level tools. The seven modes are summarized as follows:

- RUN MODE
  This is the mode in which animations are viewed. It provides menu tools for starting and stopping an animation, changing views, controlling viewing speed, jumping ahead and back in time, and more.

- DEBUG MODE
  This mode provides tools for stepping through an animation by individual events or time commands and examining the resulting movement. Information pertaining to an individual object can be obtained by clicking on the object with the mouse.

- DRAW MODE
  This mode contains the drawing tools used for creating the layout background. Tools are provided for drawing static elements such as lines, arcs, text, fills, etc. Dynamic elements such as messages, bars, plots, and layout objects are defined in Draw Mode.

- CLASS MODE
  This mode is used for defining object classes. A class serves as a template for creating both the dynamic objects that move around a layout and layout objects that generally remain stationary. The template determines an object's size and shape and other initial properties such as default speed and color. An animation will usually contain multiple object classes. For example, an animation of a hospital might contain classes that represent doctors, nurses, equipment, and patients.

- PATH MODE
  This mode is used for defining fixed paths. A path is a perfect mechanism for describing for guided, directional movement such as conveyors. The geometry or route of a path is easily defined by clicking on existing lines and arcs of a layout. Tools are also provided for defining path speeds, circularity, and accumulation status. Accumulating paths provide automatic queuing for objects that pile up at the end of the path.

- PRESENTATION MODE
  This mode is used for running scripted presentations. Scripts can include static bitmap slides and snippets of animation, separated by special effect segues such as screen fades and dissolves.

- SETUP MODE
  This mode is used for examining and altering infrequently changed configuration data. For example, the color palette can be customized or the mouse speed changed in this mode.

5 CREATING ANIMATIONS AND PRESENTATIONS

5.1 Drawing the Layout

The first step in developing an animation is to draw a layout. If a .DXF formatted CAD drawing of the system is available, a user can begin by importing the drawing into a Proof Animation layout file. This is done using the built-in CAD import/export utility. Once imported, the drawing can be examined layer by layer or by line styles. Specific layers and line styles can be deleted from the drawing as desired. When you save the resulting drawing, it is saved as a Proof Animation layout file. The original .DXF file remains intact.

If a user does not have a CAD drawing or prefers to draw using a computer, the drawing tools provided in Draw Mode are easy to use. Although it is mouse-oriented, Draw Mode also allows keyboard input, so if a user needs to draw a line of a specific length at exactly a certain angle, he or she can enter these specifications *numerically*, and the geometry will appear on the screen. To help in drawing scaled, accurate layouts, a visible grid is turned on automatically when Draw Mode is entered. For additional aid in drawing, Proof Animation has the Snap-to-Grid option. This option is also *on* as the default setting. Snap-to-Grid limits the drawing of layout elements from grid point to grid point, thus eliminating the chance of small gaps between the

endpoints of *seemingly* connected lines. Other snap options which help draw accurate layouts are Snap-to-Endpoint which *magnetically* attracts the mouse cursor to the ends of lines and arcs, and Snap-to-Tangent which quickly finds points of tangency between lines and arcs. All of these options can be turned on or off by the user during the drawing session.

5.2 Defining Object Classes

Once the background of the animation is drawn, the second step in developing an animation is usually to define one or more object classes. This is done in Class Mode. Objects and object classes are among the most important constructs in Proof Animation. A class provides the geometric description of the individual objects that move throughout the animation. The class definition also includes the initial properties such as physical clearances, color, and speed of the individual objects. Each animation will usually have a collection of object classes.

It is helpful to think of an object class as the template from which the individual objects are made. An individual object is based on the single geometric description of a particular object class. There can be an arbitrary number of *objects*, such as widgets, in the system at once, but there need be only *one* widget object class.

Motion and color-changing commands in the trace file operate on *objects*. The drawn background components, produced in Draw Mode, cannot be moved or changed. If dynamic changes in background elements are required, the appropriate components must first be defined as object classes and can then be created and positioned directly in Draw Mode. Objects that are created and placed in the layout while the user is drawing the background are called *layout objects*. Layout objects enable a user to scale and position the objects into the layout while having the background components visible as reference points. While the animation is running, layout objects can be manipulated using trace file commands. For example, if an idle machine is shown as green and a busy machine as red, the machine must first be defined as an object class. Objects from that class can be created and placed as part of the layout file, and their color can be changed while the animation is executing.

5.3 Defining Paths

Proof Animation provides two kinds of motion: absolute and guided. Absolute motion, specified by the MOVE trace file command, causes an object to be moved between two points. Guided motion always occurs along a fixed route, called a path. If objects will follow guided motion, such as travel on conveyors or along guide wires, the next step in the animation development is to use Path Mode to define one or more paths.

Paths are comprised of lines and arcs that represent the route that the objects will follow. This underlying geometry must first be drawn using Draw Mode or imported from CAD. The logical path segments are then defined *on top* of the existing lines and arcs. A single line or arc can be part of one or more paths. Once defined, paths are saved as part of the layout file.

Using paths is very simple because Proof Animation does all the work. The most commonly used trace file path command is PLACE *object*ID ON *path*. Once an object is placed on a path, it will follow that path until it visually comes to rest at the end of the path or until it is PLACEd elsewhere or DESTROYed. All objects traveling on the same path can be stopped simultaneously and resume movement at a later time. Paths provide outstanding animation power in response to a single trace file command.

Accumulating paths provide even greater power for animating paths on which queuing can take place. On accumulating paths, Proof Animation reflects physical reality by *visually* queuing objects when bottlenecks occur. This often makes a simulation model of the system much simpler to construct, because such queuing need not always be explicitly represented in the model code. Most systems contain some accumulation. This property can be used to represent certain types of conveyors, cars at a traffic signal, bank lines, and more. Paths play an especially important part in transportation, product flow, and material handling animations.

5.4 Writing the Trace File

The next step in the animation development is producing the animation trace file. Trace files consist of very readable ASCII commands. Trace files are time ordered. That means the specific animation events take place between TIME commands. Consider the following portion of a trace file:

```
TIME 34.6
CREATE PLANE 1
PLACE 1 ON RUNWAY3
SET 1 SPEED 75
TIME 52.8
```

It is very easy to visualize the results of these commands. At time 34.6, an object with an id number of 1 is created with geometry and properties inherited from class PLANE. This object will appear on screen at the beginning of a path named RUNWAY3 and begin moving along the path. The speed at which object 1

will move is set to 75 units of distance per unit of simulated time. These units are user-determined, e.g. feet and seconds. Proof Animation will continue reading trace file commands until it reads the TIME 52.8 command, signaling the end of the events that are to begin at time 34.6. It is very easy to produce simple trace files with any ASCII editor.

For most applications, it is impractical to create trace files by hand. Using a simulation model or program to generate the trace file is usually the *only* viable approach. In order to produce a trace file, output statements are inserted into the simulation model to write the appropriately formatted commands. The Proof Animation trace file command set has been designed to be easily generated. Any language with the ability to write a formatted ASCII file is capable of producing a trace file.

5.5 Building a Presentation

As an optional final step, a professional looking presentation can be built using Proof Animation. Presentation Mode lets users display scripted sequences consisting of bit-mapped screen images or slides, full animations, and/or segments selected from full animations. These presentation elements can be linked together using fades, dissolves, and other special effects, to produce a polished presentation. This is done by writing a simple ASCII presentation script file. Complete presentations can be viewed without ever exiting Proof Animation. This eliminates the awkwardness of switching back and forth between the computer and other display media during a presentation.

Slides can be created directly in Proof Animation or any software package capable of exporting industry-standard .PCX image files. There are many such packages available, and virtually all of them can produce very high-quality charts, graphs, and slides. Proof Animation can both read and write these .PCX screen images. It is very straightforward to save Proof Animation screen images as .PCX files and incorporate them into presentations as slides.

Presentations can be developed so that slides and animations appear on the screen for a defined amount of time. The viewer does not have to interact with the computer for the presentation to continue. Presentations can also be developed to continue once a key or mouse button is pressed, giving the viewer or presenter ample time to comment on what is currently on the screen.

When developing the presentation, a user can choose to highlight areas of interest within the animation by using different views or sound to draw the viewer's attention to particular aspects of the animation. Presentations can incorporate selectable menus defined by the presentation developer. These menus can be set up by topic, giving the viewer or presenter complete control and flexibility of what to show.

6 THE PROOF PROFESSIONAL ADVANTAGE

Proof Professional offers obvious advantages because it is limited only by the total memory available on a computer. No artificial memory limits are imposed; therefore, large-scale animations can be run on the PC.

Proof Professional also offers two additional versions at no additional cost: High-Resolution and High-Color Proof. Each has its own unique features.

6.1 High-Resolution Proof

High-Resolution Proof is a 4-bit color version with 7 foreground and 2 background colors that runs in 1024x768 resolution. Like Proof Professional, it imposes no memory limits. The high resolution picks up minute details in the layout and the objects of an animation. It is ideal for presentations, especially when displayed on large monitors.

6.2 High-Color Proof

High-Color has 640x480 resolution which is higher than the 640x350 standard versions of Proof. High-Color, with its 8-bit color, offers the user a unique set of features.

The most noticeable feature is the expanded color palette containing 24 foreground colors and 8 background colors. The background colors consist of 7 layout colors and 1 backdrop color. The background colors do not interfere with the foreground colors and therefore give a user much more flexibility when drawing the static background portion of the layout. With more colors from which to choose, the background can be drawn with more detail without sacrificing the color integrity.

Another major feature of High-Color Proof is the ability to draw multi-colored object classes. This enhances the animation, especially for systems that have compound objects that flow through the processes. With multi-colored objects, Proof Animation gives users the one advantage that was once limited to pixel-based animation, multi-colored objects, without all of the limitations that pixel-based animators possess.

6.3 Hardware

Although Proof Professional runs on 386-based machines with a math coprocessor and VGA display,

high-end 486 and higher PCs are ideal platforms for running large-scale animations.

In addition to the basic configuration mentioned above, High-Resolution Proof requires a 1024x768 non-interlaced monitor. Both High-Resolution and High-Color Proof require a Super VGA card that is based on a supported chip set. Contact Wolverine for details on the supported Super VGA cards.

7 THE ADVANCED FEATURE SET

Advanced features make Proof Animation unparalleled in its capabilities. Keeping the user in mind, many of these features were added because of direct requests from our customers. This feature set is described below:

7.1 Multi-Window Display

The animation screen can be divided into separate windows. Within each window, the view can be independently manipulated using zooms, pans and rotations to include a portion or all of the animation canvas.

With this feature users can maintain a window that keeps updating statistics in constant view while panning and zooming to different areas of the layout. For example, a user may choose to create a layout object that represents a clock to display the current system time and place it in a separate window that is always in view.

7.2 Displaying Dynamic Data

Along with bars, dynamic plots can be incorporated into a layout as a way of displaying statistics. A plot can be used to show any type of data displayed on X-Y axes. Plots, like bars, are defined and placed in the layout using Draw Mode. The data are displayed in the form of line segments that can be added, erased or changed via the new PLOT trace file command. A single plot can have many different types of data plotted simultaneously with each curve plotted in a different color. Plots offer a unique way of displaying changes over time. A viewer can look at a plot and see the differences in the value of the statistic as the animation progresses.

Textual messages can be incorporated into object classes. Each object created from that class will carry its own messages. The text displayed in the messages can be changed using the WRITE trace file command. Messages can be used to visually differentiate objects that are otherwise identical in appearance.

7.3 Features Affecting Object Movement

When an object is placed on a path, that object is positioned on the path at the object's hotpoint. The hotpoint of every object class is the point (0,0) as viewed in Class Mode. Placement of the hotpoint within the class is determined by how the geometry is drawn around that point. For example, the class can be drawn so that (0,0) is in the center, on an edge, or even outside of the geometry. Unlike compact objects, long, thin objects rounding corners of paths can exhibit a *fishtailing* effect, especially when the hotpoint is near the leading edge of the object, as shown in Figure 1.

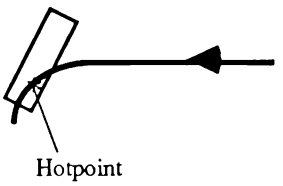

Figure 1: Hotpoint Attachment Only

Proof Animation provides the ability to supply a second point of attachment to object classes. This point is called the rear guide point or RGP. The optional RGP is defined as a property of a class in Class Mode. It is input as a negative number which serves as the RGP's displacement from the hotpoint. Both the RGP and hotpoint will remain connected to a given path as long as such connections are physically feasible. Figure 2 illustrates this concept.

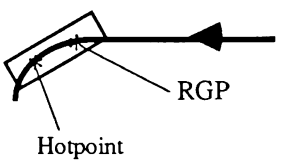

Figure 2: Hotpoint and RGP Attachment

Another feature that can affect the movement of objects is the ability to specify a negative speed or travel time for objects traveling on non-accumulating paths. Specifying a negative speed causes the object to travel in the reverse direction on a given path. This gives users the ability to have an object back into an area and pull out or vice versa. This is especially useful in animations that contain traffic of any type.

The PLACE...ON commands for objects entering paths has a new option called SQUEEZE. When SQUEEZE is used with a PLACE...ON, all objects on the path *behind* the point at which the new object will enter are pushed back, allowing sufficient room for the new object. Objects can be SQUEEZEd at the end or at any offset of a path as well as before or after a particular

object already residing on the path. This feature is ideal for animating conveyors that are loaded manually.

The addition of the ATTACH trace file command lets the user connect objects to one another. All the objects then follow the movement of the leader object. There is a single level of attachment. For example, Object 2 can be attached to Object 1, but Object 3 cannot also be attached to Object 1 while Object 2 is still attached. To create a string of objects, a user can simply attach 2 to 1 and then attach 3 to 2, and so on. The movement of all of the objects is controlled by the first object in the string, in this case, Object 1. The trace file commands need only manipulate the leader object, and all the other attached objects will follow. Speed is also determined by the leader of the chain of objects. This new feature makes it simple to animate multi-car trains or vehicles in tow. The DETACH command is used to break the chain of objects at any point of attachment.

7.4 Features That Enhance The Trace File

A JUMP option has been added to the TIME command. If JUMP is used, the trace file is automatically fast-forwarded until a TIME command is encountered with a value that is equal to or greater than the value specified after the JUMP keyword. For example, a TIME JUMP 100 command issued in the trace file will fast forward the animation until it executed a TIME 100 or higher TIME command. The animation then begins to run at the specified viewing speed, processing trace file commands normally. This is a way to skip large warm-up periods in an animation without needing to stop and enter a fast-forward time via the Run Mode menu.

There is an option in the file menu that allows the writing of an *abridged* form of the trace file. When creating an abridged trace file, a full version of the trace file must first exist. A user determines the start and end times of the trace file that will be incorporated into the abridged version. This is especially useful in preparing animations for presentations. The abridged trace file contains the information needed to immediately update the state of the animation to the start time specified in the abridged trace file and continue running normally until the user-specified end time is reached. The abridged trace file eliminates any delays that would be incurred because of the additional trace file commands that are executed during a standard fast-forward.

8 SUMMARY

Wolverine Software's Proof Animation has set the standard for maximum power and performance. Some of Proof Animation's many unique features include the ability to show statistics using bar graphs and plots, create presentations, built-in CAD import/export, drawing tools, smooth motion, and a unique multi-windowing display.

Proof Animation is not tied to a specific application. There are features that make it an ideal choice for the animation of systems like computer networks, health care, transportation, reengineering, manufacturing, and more while maintaining ease of use.

An animation benefits a user in every phase of the study: verification, validation, presentation of results, and the overall system design process. Proof Animation's unmatched features make it the perfect tool for each of these phases regardless of the application.

REFERENCES

Earle, N.J. and Henriksen, J.O. 1994. Proof animation: reaching new heights in animation. *Proceedings of the 1994 Winter Simulation Conference*, eds. J. Tew, S. Manivannan, D. Sadowski, and A. Seila, 509-516 Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Wolverine Software. 1992. *Using Proof Animation.* Annandale, Virginia: Wolverine Software Corporation.

AUTHOR BIOGRAPHIES

NANCY J. EARLE is a Senior Industrial Engineer at Wolverine Software Corporation. She received B.S. and M.S. degrees in Industrial Engineering from Purdue University, with a concentration in simulation. Her responsibilities include consulting, training, technical support, and product development support. Ms. Earle served as the Exhibits Chair for the 1992 Winter Simulation Conference.

JAMES O. HENRIKSEN is the President of Wolverine Software Corporation. He is a frequent contributor to the literature on simulation. Mr. Henriksen served as the Business Chairman of the 1981 Winter Simulation Conference and as the General Chairman of the 1986 Winter Simulation Conference. He has also served on the Board of Directors of the conference as the ACM/SIGSIM representative.