

PROOF ANIMATION: REACHING NEW HEIGHTS IN ANIMATION

Nancy J. Earle
James O. Henriksen

Wolverine Software Corporation
7617 Little River Turnpike
Annandale, Virginia 22003

ABSTRACT

The Proof Animation™ 2.0 family of products is the latest achievement in Wolverine's ongoing commitment to produce innovative animation software. Proof Animation provides our customers with software capable of animating a vast array of applications while remaining easy to use. This product family runs on readily available, inexpensive PC hardware. Proof Animation is ASCII file-driven and features an open architecture, general purpose design, vector-based geometry with the ability to zoom in or out while maintaining crisp clear images, post-processing for maximum performance, built-in drawing tools for ease of creating animations, dynamic bar graphs and plots used to display 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. Proof Animation's user interface is menu-based and easily navigated using either a mouse or keyboard. Its superior performance assures smooth, realistic motion for animations of all sizes and degrees of complexity.

1 INTRODUCTION

Nothing is more frustrating to an analyst than to have worked hard to create a valid and representative simulation model, only to find it necessary to change the model logic to make the animation look realistic. Furthermore, 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. With the introduction of Proof Animation 2.0 we have added an advanced feature set that meets the challenges of animating today's complex systems while keeping ease of use a top priority.

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 the case of manufacturing, there are many areas that benefit from simulation and animation. These areas range from staffing and scheduling to material handling. Some application-specific animation software may not be able to handle the full range of problems even within a specific application.

A successful tool should apply to *every* type of problem that will be studied. Why should organizations pay for multiple single application software packages when a general purpose animation package will meet all of their needs? Smart organizations will seek a good general purpose tool that will help them visualize the results of all types of scenarios. Proof Animation *is* that general purpose tool. It is not tied to a specific area of application. Proof Animation can be used to animate the full range of applications from new areas like Business Process Reengineering, to the classic applications like healthcare, manufacturing, traffic, and communications.

Proof Animation can be paired with most simulation and programming software. The analyst can use the modeling tool with which he or she is most experienced and comfortable and not have to settle for the limitations of animation, if any, tied to that specific modeling tool.

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.

Proof Animation can be used in every phase of the simulation study. It can help verify the model during the development stage, and it can effectively show the results of a simulation study to an audience with varying technical backgrounds using the built-in presentation capabilities. Polished presentations consisting of a collection of slides and animation clips can easily be assembled to summarize the results of the simulation project. There is even a Run-time version for users who

want to provide their clients with the ability to run different scenarios on pre-developed animations.

The Proof Animation family of products can meet the challenges of animating all types and sizes of systems. In the following sections we describe the family, discuss their design, describe their organization, give an overview of how they are used, and describe the new advanced features of Proof Animation 2.0.

2 THE PROOF ANIMATION FAMILY

The Proof Animation family runs on inexpensive PC hardware. All versions of Proof Animation run as 32-bit applications that require only a 386 or better CPU, a math coprocessor, and at least a VGA-compatible video card. Configurations of this type are very widely available, and therefore maximize portability of the animations. It is 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 both Windows 3.1x and OS/2 2.x. The following products comprise the Proof Animation family:

- **PROOF ANIMATION**
Proof Animation is the basic animator. Memory limitations apply.
- **PROOF PROFESSIONAL**
Proof Professional utilizes extended memory for animating larger systems. Included at no additional cost are our 1024 X 768 high-resolution version and our new High-Color Proof Animation that runs in 640 X 480 resolution.
- **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.

- **CAD TRANSLATOR**

This add-on utility is used for converting industry-standard .DXF CAD files into Proof Animation layout (.LAY) files. It can also be used for converting Proof Animation layout files into .DXF files.

3 THE DESIGN PHILOSOPHY

3.1 The General-Purpose Approach

Proof Animation can be classified as a general-purpose animation product because it is independent of any particular simulation or programming language. While built to work easily with Wolverine's GPSS/H simulation software, Proof Animation also provides affordable animation software to users who develop models in other popular 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, both the simulation and animation software are sold only as a pair, so both must be purchased regardless of the needs of the user. The supposed advantage of the coupled approach is that because the animator has direct access to the simulation events, development of the animation is simplified. However, the user 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 acquisition of software is that the selection can be based on optimal functionality and prices.

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 (.LAY) file and the trace (.ATF) 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 which 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, MOVE, SET SPEED, SET COLOR and many more. This file provides Proof Animation with information on when, where, and what to create, destroy, place, move, 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 which 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. That means 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 resources are 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 and to speed up or slow down the viewing speed. These features make it easy to identify and investigate unusual system behavior.

3.2 Vector-Based Geometry

In the Proof Animation 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 formatted files. In many cases, a CAD drawing already exists for the system which will be animated. If that is the case, the effort of redrawing an entire layout can be avoided. Proof Animation's optional add-on CAD Translator provides the capability of converting industry-standard .DXF CAD 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.

Instead of vector geometry, many animation packages use a pixel-oriented approach for drawing. The advantage of this approach is purely aesthetic. With the ability to manipulate individual pixels, one can produce detailed, arty 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 High Performance

Proof Animation was designed to achieve high performance. High performance enables Proof Animation to maintain very 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. This allows the audience to easily gain confidence in the results of the simulation study. Objects which move smoothly across the screen are obviously more realistic than those that appear to jump across the screen.

4 THE PROOF ANIMATION MODES

Proof Animation is organized into seven menu-driven *modes*. Each mode is a collection of closely related functions. Switching among 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 is a variant of Run mode. It 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 CAD-like 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 application for guided, directional object 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 which pile up at the end of the path.

- PRESENTATION MODE

This mode is used for running scripted presentations. Scripts can include static 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 increased or decreased in this mode.

5 USING PROOF ANIMATION: AN OVERVIEW

5.1 Drawing a Layout

The first step in developing an animation is to draw a layout. If a .DXF formatted CAD drawing of the system that will be animated is available, the user can begin by importing the drawing into a Proof Animation layout file. This is done using the CAD Translator add-on utility. The entire CAD drawing or selected layers can be imported. If the user does not have a CAD drawing or prefers to draw using a computer, the drawing tools provided in Draw Mode are easy to learn and use. Although it is mouse-oriented, Draw Mode also allows keyboard input, so if the 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 element from grid point to grid point, thus eliminating the chance of small gaps between the endpoints of *seemingly* connected lines. There are other snap options to help draw accurate layouts. They 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 an optional speed of the individual objects. Each animation will usually have a collection of object classes. For example, an animation of a job shop might include object classes for a person, machine, forklift, and pallet.

It is helpful to think of an object class as the template from which the individual objects are made. Expanding on the animation mentioned above, machines can display different colors when busy and idle. Each of the machines is an individual object based on the single geometric description of the same class. There can be an arbitrary number of machine *objects* in the system at once, but there need be only *one* machine object class.

Motion and color-changing commands in the trace file operate on objects. Most layouts are drawn directly using Draw Mode. The drawn background components cannot move or change color. If such changes 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*. This enables scaling and positioning of the objects 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

The next step in developing an animation is to use Path Mode to define one or more paths, if guided motion is a part of the system. 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 user-defined points. Guided motion always occurs along a fixed route, called a path. The lines and arcs that comprise the path 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 objectID 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. A surprising number of 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 Producing a Trace File

The next step in the animation development is writing the animation trace file. Trace files consist of very readable 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 0
CREATE AUTO 1
SET 1 COLOR BLUE
PLACE 1 ON PATH1
```

It is very easy to understand and visualize the results of these commands. At time 0, an object with an id number of 1 is created with geometry and properties inherited from class AUTO. This object will appear on screen colored blue and be placed at the beginning of a path named PATH1. 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 produce the appropriately formatted commands. The Proof Animation trace file command set has been designed to be easily generated. Any language that can do formatted writes to an ASCII file is capable of building a trace file.

5.5 Building a Presentation

As an optional final step, a 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 (.PSF). Complete presentations can be viewed without ever exiting Proof Animation. This eliminates the awkwardness of switching back and forth between overhead transparencies and computer displays during a presentation.

Slides can be created directly in Proof Animation or using third party packages 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, the user can choose to highlight areas of interest in both space and time 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 the 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 the 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 big 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 the users the one advantage that was once limited to pixel-based animation, detailed and arty 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 based PCs are ideal platforms for running large-scale animations and are available for reasonable prices.

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 SuperVGA card that is based on a supported chip set. Contact Wolverine for details on the supported SuperVGA cards.

7 PROOF 2.0 ADVANCED FEATURES

Proof Animation 2.0 was developed keeping the user in mind. Many of the new features were added because of direct requests from our customers. This new feature set, described below, makes Proof Animation unparalleled in its capabilities.

7.1 Multi-Window Display

The animation screen can be divided into separate windows. Each window can be independently manipulated using zooms, pans and rotation to provide a unique view of 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. The user may also choose to create a layout object that represents a clock that displays the current system time and place it in a window that is always in view.

7.2 Features Affecting Movement of Objects

When an object is placed on a path, that object is positioned onto 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 be outside of the geometry. Unlike compactly drawn objects, long, thin objects rounding corners of paths can exhibit a *fishtailing* effect, especially when the hotpoint is near the very front of the object, as shown in Figure 1.

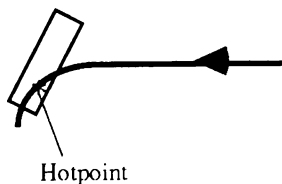


Figure 1: Hotpoint Attachment Only

Proof Animation 2.0 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.

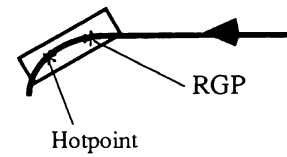


Figure 2: Hotpoint and RGP Attachment

Another new 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 the 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 the path 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 a string of objects to one another. All the objects then follow the movement of the leader object. This new feature makes animating multi-car trains or vehicles in tow a snap.

7.2 Displaying Dynamic Data

Along with bars, plots can now 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 are dynamic. Plots, like bars, are defined and placed into the layout from within Draw Mode. Data line segments of a particular plot can be added, erased or changed via the new PLOT trace file command. A single plot can have many different types of data being plotted simultaneously, each curve being 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.

Messages can be incorporated into object classes. Each object created from the class will carry its own

message. The text displayed in the message can be changed using the WRITE trace file command.

7.3 Features Affecting The Processing of Trace Files

A JUMP option has been added to the TIME command. If JUMP is used, the trace file is automatically processed in a fast-forward mode until a TIME command is encountered with a value that is equal or greater to the value specified after the JUMP keyword. For example, a TIME JUMP 100 command issued in the trace file would 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 a new 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. The 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

Simulation and animation technology is constantly improving. Wolverine Software Corporation continues to be at the forefront of this technology by providing Proof Animation 2.0, an innovative family of animation products.

Some of Proof Animation's many important features include the ability to show statistics using bar graphs and plots, create presentations, built-in CAD 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, manufacturing, and more while maintaining ease of use.

An animation benefits the user in every phase of the study: verification, validation, presentation of results, and the overall system design process. Proof

Animation's unmatched feature set and general purpose architecture makes it the perfect tool for each of these phases. Wolverine Software has once again brought animation software to new heights while keeping the ease of use and affordability down to earth.

REFERENCES

- Earle, N.J. and Henriksen, J.O. 1993. Better Animation For Your Simulation. In *Proceedings of the 1993 Winter Simulation Conference*, eds. G. Evans, M. Mollaghasemi, E. Russell, and E. Biles, 172-178. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.
- Henriksen, J.O. and N.J. Earle. 1992. Proof Animation: The General Purpose Animator. In *Proceedings of the 1992 Winter Simulation Conference*, eds. J. Swain, D. Goldsman, R. Crain, and J. Wilson, 366-370. 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.