Proceedings of the 1991 Winter Simulation Conference
Barry L. Nelson, W. David Kelton, Gordon M. Clark (eds.)

PROOF ANIMATION: THE GENERAL PURPOSE ANIMATOR

Daniel T. Brunner
Nancy J. Earle
James O. Henriksen

Wolverine Software Corporation
4115 Annandale Road
Annandale, Virginia 22003

ABSTRACT

Proof Animation™ is high quality PC-based animation and presentation software. Proof Animation runs on any 286 or better computer with a math coprocessor, EGA or VGA graphics, and a mouse. Among Proof Animation's many advanced features are smooth motion, true zooming, and presentation portability. An open architecture makes Proof Animation compatible with most popular simulation software.

1 INTRODUCTION

During all phases of a simulation project, animation is a critical tool in presenting simulation results to management and clients. Animation is also important during model building. Users of simulation software find that complex logic problems often (though not always) come to light quickly when animated. Furthermore, animation can play an important part in the overall system design process as well as in presentation and model building. In this paper we describe characteristics of Proof Animation that help simulation modelers address all of these needs.

Throughout this paper, we use the Proof Animation Release 1.0 designation wherever it is necessary to distinguish the current product from possible future versions.

2 ABOUT PROOF ANIMATION

2.1 Overview

Proof Animation Release 1.0 is a PC-based product. The minimum hardware is a 286 machine with a math coprocessor and either an EGA or VGA display. This common hardware allows Proof Animation to be portable, which is especially advantageous when projects are presented off-site.

Proof Animation Release 1.0 is a "post-processing" animation package. In order for an animation to run, two files must exist: the layout file and the trace file. Typically, a simulation model produces the trace file that drives the animation. There are tradeoffs when this approach is compared to "concurrent" animation, which animates while the simulation executes—but many of the tradeoffs favor post-processing.

Proof Animation offers a special "presentation mode." Professional-looking static slides can be created and interwoven with animation segments and shown on the computer screen or with a computer projection display. It may soon be possible for a Proof Animation user to take such a presentation and create a diskette that can be given to others for presentation viewing.

Proof Animation has an open architecture. This makes Proof Animation compatible with many popular simulation software packages. In fact, simulation-specific software is not necessarily needed to drive the animation. If the software used has file I/O that is capable of producing an ASCII file, it can produce a Proof Animation trace file.

Proof Animation has a geometry-based (CAD-like) internal data structure. Changing the orientation or scale of an object, static or moving, will not affect the quality of its appearance on screen. Many options are available for zooming, panning, rotating, or re-orienting the animation. Individual objects can rotate while they move around curves.

Although Proof Animation is not a full-featured CAD program, it does have a mouse-driven, CAD-like drawing mode. Through a series of pull-down menus, the layout geometry and path and shape definitions can be created using using the mouse. It is also possible to read existing CAD layouts into a Proof Animation layout. Proof Animation's drawing mode has been developed to easily handle a wide variety of systems. This means that diverse applications, such as network communications and health care systems, can be handled as easily as complex material handling systems.

2.2 Hardware Requirements

Proof Animation Release 1.0 runs on IBM or compatible PCs, primarily because of the large installed base of color-capable systems. The MS-DOS operating system was chosen for its large installed base and also because it behaves as a single-tasking environment (even under some third-party multitasking software). When Proof Animation can take total control of the CPU, the result is an animation with very smooth motion.

New simulation packages have been developed, such as GPSS/H 386, that take advantage of a 386- or 486-based machine's faster 32-bit architecture and large memory address space while still running under MS-DOS. This means that an MS-DOS PC can handle large simulation models. Using Proof Animation, the user can do almost any simulation and the animation on one machine.

Proof Animation Release 1.0 supports both EGA and VGA color graphics displays at a resolution of 640 x 350. In Proof Animation, one pixel needs four bits of video memory, and four times 640 times 350 (the screen dimensions, in pixels) is 896,000 bits, or 112K bytes. Given the 256K of video memory found on nearly all EGA cards and on every VGA adapter, Proof Animation can *double buffer* a 112K screen — that is, improve the appearance of the animation by keeping two copies of the screen in video memory, displaying one while updating the other. Double buffering is very important for quality animation.

We are carefully studying the evolving XGA, 8514/A, TIGA, and "Super VGA" display options, all of which offer resolution at or above 1024 by 768. Once the PC graphics market has moved clearly in one of these directions, the resolution of Proof Animation will be enhanced.

2.3 Post-Processing vs. Concurrent

Proof Animation Release 1.0 is *post-processing*, meaning the simulation runs first, then the results drive the animation. Some animation packages run *concurrently*, displaying state changes while the simulation runs. What are the tradeoffs?

Although it is possible with most concurrently running simulation/animation software to make certain limited types of changes to the system and watch the impact, many types of changes (such as scheduling algorithm changes) are difficult or impossible to animate without advance work by the modeler.

Concurrent animation is completely dependent on the execution of the simulation model. This can be very tedious when the software is under consistent use (e.g. during the model building phase), especially if the underlying simulation software is not particularly fast at executing.

With a post-processing animator, it is possible to fast forward quickly to any point in simulated time. Concurrent animators can only do this as fast as the simulation will run.

Post-processing adds to the portability of Proof Animation. The user needs to take only the Proof Animation software to a remote location, and the target machine need not be equipped to handle the simulation software. Thanks to ongoing "demo maker" development efforts, it may soon be possible to detach animations for distribution to others who don't have the main Proof Animation program at all.

Despite all of these arguments in favor of post-processing, there are those who do not believe they can do without concurrent animation. To address this requirement, we are planning a future version of Proof Animation that is capable of concurrent animation.

3 PROOF ANIMATION FEATURES

3.1 Presentation Mode

Proof Animation has a full-featured presentation mode. This lets users create "slides" made up of words, objects, screen snapshots, or even dynamic animated segments. These slides can be linked together to produce a polished, complete presentation. Awkward transitions from overhead transparencies to a computer display during a presentation are eliminated. The presentation developer can choose to highlight areas of interest (in space or time) within the animation, and thus draw the viewer's attention to particular aspects of the simulation.

3.2 Open Architecture for Trace Events

Some animation software is integrated into a simulation language or package. In order to use the animation, the user must go through the process of building a simulation model using the integrated tool. Other animators use post-processing, but the specifications of the trace file are generally not available to the user.

Proof Animation has an open architecture. The specifications for generating trace events are public and easily followed. The most dramatic impact of Proof Animation's open architecture has initially been the quick adoption of Proof Animation as the animation engine of choice by many people using simulation software other than Wolverine Software's own GPSS/H.

It is also possible to build graphical depictions of systems that have not been simulated, or to build a new simulation/animation package around the Proof Animation graphics engine. Proof Animation can also be used by non-simulationists as presentation software. The open architecture opens some wide doors for Proof Animation and its users.

The trace event architecture of Proof Animation consists of a very simple, record-oriented animation language with English-like commands. A typical Release 1.0 animation consists of a layout file and an animation trace file. The layout file contains static geometry information and definition commands. The trace file is used for recording the time-dependent information that controls all animation activity. Both files are populated with printable ASCII characters.

Here are a few examples of the easy-to-use trace event commands:

```
SET...COLOR...
MOVE
PLACE..ON..AT..
CREATE
TIME
DESTROY
```

Normally, a small set of commands used over and over comprise the animation trace file. The process of actually writing the trace file is automated. For simulation, this means that the model writes commands such as SET COLOR into the file each time an entity passes a certain point in the simulation. For non-simulation applications such as control algorithm debugging, the process can be similarly automated.

3.3 Geometry-based Graphics Data Structure

A CAD-like, vector-based structure is used in Proof Animation, allowing the software to rotate an object, pan, and zoom in or out without losing the integrity of the object. In contrast, zooming in with a pixel-based system makes the object's edges appear jagged.

The power of a CAD-like data structure provides benefits in two areas. The first is the versatility of the available drawing tools. The second is the flexibility with which the display can be manipulated. Proof Animation's CAD-like architecture allows total control of the viewing environment. This is unprecedented among PC simulation animators. The geometric data structure allows complete panning, zooming, rotating, and changes of viewpoint.

Complementing the graphics data structure is a CAD-like drawing mode for creating the layout file. Using a series of menus and a mouse, the static layout, dynamic objects, and paths can be created.

3.4 CAD Interoperability for Design

Proof Animation is the first animation software that enables a two-way CAD interface via the DXF file format. A separate utility makes it possible to read in an existing DXF file to create the background portion of the Proof Animation layout file. This saves time in developing the animation. Then, if changes are made in the animation layout, the utility can produce an updated DXF file (comprised only of the subset of CAD primitives available in Proof Animation). The project design team can now rely on the simulation and animation as a timely and dependable design check. If changes are needed, they can first be tested with the simulation. Once a final design is achieved, the updated animation will produce the final layout in a file that can then be read into nearly any PC CAD system.

3.5 Smooth Motion

Smooth motion was a primary design goal for Proof Animation, and it has been achieved with stunning success. In most media, it is necessary to create and recreate static images rapidly in order to create the illusion of motion. This is, of course, the principle behind motion pictures and television as well as cartoon animation.

In the case of a computer and raster-based CRT screen, or the equivalent raster-based video game, the image is created as a set of discrete pixels represented in video memory. For these applications, the pixel representation must be either recreated continuously at different locations, or saved and "blitted" to different locations on the screen. This process must be repeated many times per second, or the motion will appear jerky.

How fast is fast enough? Motion pictures run at about 20 frames per second, and standard television at about 30 frames. Simulation animation software available in the 1980s was plagued by slow frame rates. Due to the discreteness of the pixels and the resulting high bandwidth images, computer displays of artificially created objects can require even higher frame rates than television, or the motion will appear to buzz or jerk. The frame rates on much of the available software have been on the order of 10 frames per second or less, while Proof Animation has starting rates of 60 to 70 frames per second.

When Proof Animation cannot move pixels quickly enough to keep up with such high frame rate, the frame rate is reduced in order to maintain a constant (though user-adjustable) ratio of animated time to "wall clock" time. With other animation software, the apparent speed of objects moving across the screen generally diminishes in such circumstances. Proof Animation performs this adjustment continuously. With Proof Animation's high starting frame rates, the effect of reducing the frame rate remains visually acceptable.

4 THE LAYOUT FILE

A Proof Animation layout file contains basic geometry and also definition information. The basic geometry is simply the lines, arcs, fill points, and text that make up the static background of the animation. The definition information consists primarily of Object Class definitions, Path definitions, Object initializations, Message definitions, and Named View definitions.

A Proof Animation Layout file is generally developed or completed using the mouse and saved from Proof Animation. The format is ASCII and open, which allows programmers to develop other software which can read and possibly modify or create the contents of a Layout file. However, most users should never need to look at the ASCII contents of a Layout file

4.1 Object Classes and Objects

Among the most important constructs in Proof Animation are the Object Class and the Object.

An Object Class is a geometric description of some type of object, such as an automobile. A traffic model might have five different Object Classes: Automobiles, Trucks, Buses, Campers, and Motorcycles. In addition to shape information, an Object Class contains a few other properties such as physical clearances, color, and an optional speed.

Although Proof Animation does not purport to implement a true "object oriented" framework, it is meaningful to call an Object an "instance" of an Object Class. Expanding on the traffic model mentioned above, one could have northbound and southbound cars; cars making continuous turning movements; red, green, or beige cars; large cars and small cars. Each of these cars is an Object, based on the single geometric description of an automobile. There can be an arbitrary number of "Automobile" Objects in the system at once, but there need be only one "Automobile" Object Class.

All of the motion and color-changing primitives in Proof Animation operate on Objects. Most layouts are drawn directly on the screen, and the background geometry components cannot move or change color. However, if movement or color change is desired, then the appropriate components can be made into Objects.

4.2 Paths

The simple things the user can do with an Object include: CREATE or DESTROY, PLACE (making it visible), SET COLOR and SET SPEED, and MOVE (causing the Object to move smoothly from points A to B.) Object movement can also be achieved via a Path. A Path is a graphically defined data structure composed of references to parts of existing lines and/or arcs. Once defined, Paths are saved as part of the layout file.

The more complicated things one can do with an Object involve Paths. Actually, using Paths is very simple, because Proof Animation does all the work. The most commonly used Path command is PLACE [object] 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). Paths provide outstanding power in response to a single trace event command.

A variant is the Accumulating Path, which offers even more power. On an Accumulating Path, Proof Animation reflects physical reality by allowing objects to queue visually if there is a blockage. This often makes a simulation model of the system much simpler to construct, because such queueing need not always be explicitly represented in the model. A surprising number of systems behave in this manner, from certain types of conveyors to supermarket checkout lanes. Paths play an especially important part in transportation and material handling animations.

4.3 Static Objects

Objects that represent moving entities are usually not permanent. Such Objects are typically created in the trace event file. However, other Objects might persist throughout the animation. Their initial placement might be at a coordinate location rather than on a named Path. We call such an Object a Static Object. The layout file can save individual Objects that are created and placed using the mouse prior to running the animation.

The Static Objects feature was added to Proof Animation just before final release. It serves two important functions. First, this feature largely relieves the simulation model of needing to deal with the coordinates of resources depicted in the animation. Second, Static Objects allow Proof Animation to be used as a sort of model builder. If the relationship between specific Object Classes and corresponding simulation constructs can be well defined, then a user could simply place Static Objects on the screen and save the layout for further processing by a model generator program.

We must emphasize that Proof Animation, Release 1.0, is not a model builder. The main limitation is the inability to set up custom-defined parameters that a model-building user could modify when placing the Static Objects. However, the Static Objects approach offers possibilities for future enhancements.

4.4 Messages

A Message in Proof is a definitional construct. It looks and acts similar to regular text, but the contents of the text can be changed from the trace file. Messages are named, placed, and saved during layout construction. Properties such as character size, location, and rotational orientation are saved. Messages are used for later display of statistics or status information.

4.5 Named Views

Proof Animation supports Named Views. At any time during layout construction or with an animation running, a user can preserve the current View by attaching a name to it. A View in Proof Animation contains properties such as center point and scale factor. Once meaningful Named Views have been defined, a view can be accessed quickly from Draw Mode or Run Mode.

Proof Animation provides three pre-defined Views: Home, Class, and Previous. The Home view is used at startup. The Class view is used during Class definition only. The Previous view always returns to the previous view. The Home and Class views can be modified and re-saved.

Named Views would be unimportant in an animator that offered only one or a few screens' worth of animation. However, Proof Animations can be very large, and Named Views can be very helpful for navigating them. Example Named Views in a particular animation might include Loading Area, Stat Summary, and Work Cell 12A.

5 SUMMARY

Animation is a powerful addition to any simulation effort. An animation benefits the modeler in verification, validation, and presentation of results, and helps with the overall system design process.

Simulation and animation technology is improving. Wolverine Software Corporation is contributing to this improvement by providing an innovative animation package called Proof Animation. This general purpose animator boasts many important features. Among these features are the ability to create presentations, an open architecture (for compatibility with a variety of software), a CAD-like structure, smooth motion, and powerful drawing features.

REFERENCES

Brunner, D.T. and N.J. Earle. 1991. *Using Proof Animation.* Annandale, Virginia: Wolverine Software Corporation.

Brunner, D.T. and N.J. Earle. 1991. *Proof Animation CAD Translator User's Guide.* Annandale, Virginia: Wolverine Software Corporation.

Brunner, D.T. and J.O. Henriksen. 1989. A General Purpose Animator. In *Proceedings of the 1989 Winter Simulation Conference*, eds. E.A. MacNair, K.J. Musselman, and P. Heidelberger, 249-253. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Earle, N.J., D.T. Brunner and J.O. Henriksen. 1990. Proof: The General Purpose Animator. In *Proceedings of the 1990 Winter Simulation Conference*, eds. O. Balci, R.P. Sadowski, and R. Nance, 106-108. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

AUTHOR BIOGRAPHIES

DANIEL T. BRUNNER received a B.S. in Electrical Engineering from Purdue University in 1980, and an M.B.A. from The University of Michigan in 1986. He has been with Wolverine since 1986, where his responsibilities include product marketing, product development support, training, and simulation consulting. Mr. Brunner served as Publicity Chair for the 1988 Winter Simulation Conference and is Business Chair for the 1992 conference.

NANCY J. EARLE received B.S. (1982) and M.S. (1984) degrees in Industrial Engineering from Purdue University, where her concentration was in simulation. She joined Wolverine as an industrial engineer in 1989. Her responsibilities include consulting, training, technical support, and product development support. Previously, she worked for Corning, Incorporated as a simulation analyst in manufacturing. While there, she developed and taught short courses in simulation. Ms. Earle is a member of SCS and will serve as 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 and has presented many papers at the Winter Simulation Conference. 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 representative.