2

Chapter Two
INTRODUCTION

What is Proof Animation?

Proof Animation is a family of general-purpose system animation software for the PC.

The commercial versions of Proof include Proof Professional, Personal Proof, and Run-Time Proof

Non-commercial versions of Proof include Student Proof and the Proof Demo Viewer. The term "Proof Animation," or "Proof" for short, as used in this manual, refers to any of the versions.

The Proof Demo Maker, formerly available only as a purchased add-on feature, is now included free-of-charge with Proof Professional, Personal Proof, and Student Proof.

Proof is a vector-based, 2D animation system. It consists of:

- A CAD-like drawing capability for creating layouts and defining shapes and paths
- A set of animation commands and the ability to process program-generated sequences of these commands
- A set of presentation commands and the ability to process hand-tailored sequences of these commands

Proof is provided in two forms, as a file-driven, stand-alone program, and as a library of functions that can be called from an executing program. Most Proof users use the stand-alone version of Proof. Their simulation models write *trace files*, which are subsequently played back by Proof. This mode of operation is called *post-processed* animation. When the library versions of Proof are used, an application transmits trace commands directly to Proof, without having to use trace files. This mode of operation is called *concurrent animation.* Throughout this book, we'll almost always use the term *trace stream* to describe the stream of commands that tells Proof what to do. The command stream so described can be contained in a post-processed file or transmitted one-

at-a-time from an executing program.

Proof is called "general purpose" for two reasons. First, because it functions independently of any specific simulation software or programming language and has an openly documented command set, Proof is compatible with a wide variety of software. Second, the flexible nature of the command set allows for the animation of many types of systems in many different ways.

Proof creates precise, smoothly flowing depictions of complex systems. Proof does not create photorealistic sequences or cartoon-like, frame-by-frame effects. Most animations created with Proof portray objects moving across a static drawing, called a *layout*, that has a more schematic than artistic character. For many users, layouts are built from imported CAD drawings

Proof has the following characteristics:

- When run as a post-processor, Proof is not connected to any simulation language or tool during the course of an animation. Post-processing allows full exploitation of processing power and graphics hardware, maximizing performance. It also allows animations to run on a computer other than the one used to run the simulation, perhaps at a different location. Post-processing allows more in-depth study of long-term system behavior through rapid animation execution and fast-forwarding. Finally, and perhaps most importantly, animations can be based on models developed under a wide variety of simulation software – or even under a non-simulation programming language or other software environment.
- The Library versions of Proof offer the opportunity to visualize simulated systems as the simulations are running.
- Proof supports resolutions ranging from 640 x 480 to 1600 x 1200. 640 x 480, however, is only marginally adequate for viewing animations, and is totally unacceptable for developing animations.
- Proof allows animation over a coordinate system larger than a single screen. Any part of the layout can be viewed at any scale and without loss of resolution using Proof's zoom features. The screen can even be divided into individual windows, each of which contains an independent view of the overall coordinate system.
- Proof supports two-dimensional drawing and animation. In order to provide a quasi-3D effect, Proof provides an "isometric" (from above and off to one side) viewpoint at any rotational orientation.
- Because of its CAD-like orientation, Proof facilitates *and encourages* "to-scale" visual depictions of complex systems.
- Proof provides a steady (though adjustable) ratio of animated time to "wall clock" time. Variation in the number of objects in motion does not cause the animation to speed up or

to slow down. Thus, Proof also facilitates "to-scale" perception of the passing of simulated time.

Proof is driven by ASCII commands that can be easily generated manually by people or automatically by software. Manual generation of trace commands is not practical for real animations, although we use it in this book to teach the command set. Most users use modeling or programming software to automatically create the sequences of animation commands.

It is also theoretically possible to create layout files by hand or with a program, but most users use Proof's drawing capabilities for creating layouts.

Developing Animations

Since you're reading this book, you probably plan to *develop* animations and presentations. The things you need to learn for the work you need to do, while not too demanding, are more involved than the effort of someone who only needs to *run* the animations and presentations created by you. This manual is for the animation developer. Although "how to run an animation" is described in Chapter 4, most people who view your animations should have little trouble running an existing animation or presentation without any documentation.

As animation developer, you'll find Proof useful in the modeling process for *debugging* your simulation model or program, and for *verifying* that your model behaves correctly. You will also find Proof useful for *showing* others your work to help make sure your assumptions are valid, and for *selling* the insights gleaned from building and running the underlying model.

As a developer, you can also create complete *presentations* to any degree of sophistication. These presentations can include static slides intermixed with clips of running animations. You can run a presentation on a projection-equipped computer for large groups (see Chapter 19) or on a normal desktop computer for small groups.

You can easily transport the animation environment to another computer if necessary. Your audience can run a presentation in self-paced fashion, even if they know little or nothing about Proof. In fact, the "Demo Maker" feature allows you to distribute completed animations or presentations to others without paying any royalties.

Uses of Proof Animation

Despite its simple structure, Proof can be used in many different ways. Here are some applications that Proof has been used to animate:

- High Speed Conveyor Systems
- Urban Mass Transit Systems (Rail)
- Flexible Manufacturing Systems
- Open Pit Mines
- Robotic Handling Systems
- Street and Highway Traffic
- Just-In-Time Manufacturing
- Voice Communications Processing
- Hospital Emergency Rooms
- Paint Mixing Systems
- Automotive Transfer Lines
- Large Scale Sortation Systems
- Computer Operating System Performance
- Semiconductor Fabrication
- Material Handling System Sales Presentations
- Large Scale On-line Computerized Transaction Processing
- Distribution Centers
- Automated Guided Vehicle Systems
- Bulk Cargo Loading/Unloading Facilities
- Hospital Radiology Departments

Philosophy of Animation

A simulation model is an abstract representation of a real or contemplated system. When you develop models of systems, you must decide which properties of the system are important enough to be included in your models. You usually cannot afford to include every last detail; otherwise, model development time will be too long. Similarly, an animation is a representation of a model. Usually you cannot afford to include every detail of a model in an animation. Proof allows you to choose the amount of detail to animate.

The essential challenge in developing an animation is to select which features are most important and how best to exploit a visual medium for portraying these features. For some systems, exploiting the visual medium is straightforward. For example, when animating a factory floor, you can quickly decide which equipment and people you want to see moving around in an animation. On the other hand, if you're animating a telecommunications system, you must be much more clever in your exploitation of the visual medium. You cannot animate individual packets of information moving over a communications link. (There are too many of them, and they move too fast.) We can, however, aggregate packets into little "blobs," and allow each blob to represent 10,000 packets and move the blobs at 1 / 10,000 of their real speed. In addition, we can use colors and "dancing" bar graphs to indicate traffic intensity.

Proof is equally capable of animating a factory floor and a telecommunications system. The animation software simply moves objects over a canvas, in accordance with well-defined rules. The burden of creatively exploiting Proof's capabilities in such diverse applications rests on the shoulders of the animation developer

Conventions Used in this Documentation

References are shown in **boldface** to specify Proof menu commands and dialog box options. References are also capitalized to show unique elements of Proof, e.g. Line, Object, Path, Class.

Animation and Presentation Commands are always shown in some form of sans serif typeface. Command keywords are printed in **sans serif boldface** in the syntax examples but in the body of the text are in sans serif without the boldface, because the frequent use of boldface can create confusion. Variable components in the command syntax are always in *sans serif italics*.

File names and folder names are also shown in sans serif. Thus you'll see "Open the sample layout named sample.lay from the sample folder."

The "|" separator is used to indicate that exactly one option must be chosen from a list of two or more options so separated. Square brackets [ ] are used to enclose optional components. Curly brackets { } are used to indicate grouping in certain cases where | and [ ] might otherwise cause confusion. For example,

shout { hey [you] } | stop

specifies that shout must be followed by hey, hey you, or stop. So, the allowable forms described by the syntax above are:

shout hey

shout hey you

shout stop

Note: although some syntax examples in this book take up more than one line of text due to lack of space on the page, please notice that commands that are typed into a Proof text file must fit onto one line.

Points that need special emphasis are printed in a box in all italics, like this.

About the Examples and Exercises in this Book

Numerous examples and exercises are used throughout this book to illustrate specific points, engaging reader participation wherever possible.

Files are stored in the exercise, sample, and features folders.

All Wolverine-provided files needed to complete Student Proof exercises are located in the exercise folder.

We recommend that you create your own files where indicated in the exercises. However, if you get stuck, you can find completed versions of many of the user-created files in the sample folder.

Limitations of Student Proof Animation

The Student Proof software contains the following limitations:

- The Animation Trace File processing ends after 1,600 lines are read.
- Saved Layout Files larger than 17,500 bytes cannot be read at all.
- Execution stops after 120 seconds (two minutes) of animated running time.

Other versions of Proof do not have these limitations. Student Proof can be used in place of the Proof Demo Viewer. It automatically detects files created by the Proof Demo Maker and waives the usual limits imposed.