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

AUTOVIEW

Richard Stafford

AutoSimulations, Inc.
655 Medical Drive
Bountiful, Utah 84010, U.S.A.

ABSTRACT

This presentation is a tutorial on *AutoView*, an *AutoMod* extension package that recreates model animation according to a user-defined script. *AutoView* allows users to restart animation and move back and forth in time and 3-D space. Animation is created using text files generated from an *AutoMod* simulation model. By creating a user-defined camera description file, animation can take place that allows for panning from one view to another, as well as for attaching the camera to a simulated load or vehicle and traveling with that entity during portions of the animation. The animation is provided at an enhanced response time because it does not have the logical and statistical processes of *AutoMod* (which can slow animation). This tutorial demonstrates several animation scripts previously constructed with *AutoView*, discusses the creation of *AutoView* files from *AutoMod*, and describes the development of a camera description file.

1 INTRODUCTION

You may have heard the saying, "animation is just a pretty picture." Yet most successful simulation packages have provided some level of animation ability. There are several reasons why animation assists in making simulation software packages successful:

a. Animation provides a useful verification tool to insure that the model is properly simulating reality. Insuring model accuracy can be difficult, and sometimes impossible, unless animation is used.

b. Animation is often a useful analysis tool. Isolating model or system problems visually is much easier than stepping through code or pages of model output.

c. Animation provides a method of transferring technology and concepts to management. The saying "a picture is worth a thousand words" is very true when it comes to presenting simulation results to management. Providing managers with an animated film of a new or redesigned facility provides them with a concrete image of their operations. This is often necessary before management appropriates funds for a project.

d. Animation is often used to train operators with new or refit equipment. Operators make hundreds of logistic decisions a day. With animation, operators can also visualize the effects their decisions have on the system. Most often this method of training is more beneficial than training using the actual operation. With simulation, operators get a global view of the impact made on other departments when making operational decisions. This is not possible in a real operation because operators have a very localized view of the entire facility.

AutoView is an *AutoMod* extension that recreates model animation according to a user-defined script. It is a post-processed animation tool that provides for enhanced response time, because it does not have the logical and statistical processes of a simulation model that typically slow animation. Animation is created using text files, which are most often generated from an *AutoMod* simulation model. Since it is a post-animation tool, *AutoView* allows users to restart animation and move back and forth in time and 3-D space. Creating a user-defined description file allows for animated panning from one view to another and allows the user to attach the camera to a simulated load or vehicle, and then travel with that entity during portions of the animation. *AutoView* also allows the user to view an existing model from any angle and for any length of time.

The following sections discuss the seamless integration of *AutoMod* and *AutoView*, describe the *AutoView* user interface, explain *AutoView* viewing and animation capabilities, and construct a typical camera description file.

2 CREATING AUTOVIEW FILES

AutoView files are most often generated from an *AutoMod* model run, but this is not an *AutoView* requirement. *AutoView* reads standard ASCII-text files with structured syntax; thus, any program can write

AutoView files. *AutoMod* provides a seamless integration with *AutoView*.

Any *AutoMod* model may be used to create *AutoView* graphic files by creating graphic files during any run control snap of a simulation run. The term *snap* is derived from the word snapshot, meaning the picture or state of a model at a given instant. A snap is a period of simulated time during which all, some, or none of the following events may occur:

a. *AutoView* graphic files are created.

b. An *AutoMod* report is generated.

c. Statistics, collected internally by *AutoMod*, are reset.

d. A detailed event trace of the snap period is created.

e. *AutoMod* business graphs are plotted.

The snap length, or period of time for each snap, which is also defined in the Snap Control window, should be kept as short as possible because the generated files can become quite large. When invoked, *AutoView* reads all of the required files into Random Access Memory (RAM). Thus, the user may jump to any point in simulated time without waiting for hard disk access. The size of these graphic movement files is determined by the number of objects moving through the model and by the snap length. The best way to determine the potential size of a movement file is to run the model with a short *AutoView* snap (one minute) during a typical activity period, then check the file size. The size of the files that can be processed depends on the amount of available RAM.

Users can reduce the size of their movement files by being selective about when movement records are written. For example, if nothing of interest happens in the model until two and one-half hours into the model run, we can run the model with two snaps:

a. Make the first snap two and one-half hours long, but do not write to *AutoView*.

b. Make the second snap as short as possible, and write to *AutoView*.

Once the user has defined snaps, the *AutoMod* model can be compiled and run. *AutoView* files are created automatically when the model is run. The directory that is created is named <modelname>.avw. The following files are created during an *AutoMod* run and placed in the <modelname>.avw directory.

<system>.sdf An sdf file or 'system description file' is generated for each system in the model. This file describes the type of system, flags for colors of entities in system, what should be displayed, and other system-specific data.

<system>.odf An odf file, or 'object description file,' is generated for each movement system in the model. This file describes objects, such as vehicles, that are found in the movement system.

<system>.pdf A pdf file, or 'path description file,' is generated for each path-type movement system in the model, such as a conveyor and vehicle system. This file describes the paths available to loads and vehicles.

camera.des The *AutoView* model can have only one camera.des file. This file describes the viewing script for the model animation.

<model>.mdf The *AutoMod* model generates an mdf or 'motion description file'. This file describes the motion for all loads and vehicles.

<model>.vdf A vdf file, or 'view description file' is created by *AutoView*. The file contains the views created for the animation.

3 AUTOVIEW USER INTERFACE

Once *AutoView* graphic files have been generated, *AutoView* can be started. The *AutoView* interface is easy to use.

To start *AutoView*, type "aview <modelname>". The *AutoView* Simulation window appears. The *AutoView* environment is similar to the *AutoMod* Simulation Environment: the Simulation window appears with the Measurement Control, Grid Control, View Control, and Plotter icons. Several menu options are available:

Display The 'display' option turns on, turns off, or changes the color of elements in each system.

Edit Script The 'edit script' option is used to edit the camera.des script file.

Do/Cancel Script The 'do script' and 'cancel script' menu options toggle to start or stop execution of the user-defined scripts.

Quit The 'quit' option closes the current animation and exits *AutoView*.

When we access *AutoView*, the model appears exactly the same way it would appear in *AutoMod*. In addition, the model animation behaves exactly as it does in *AutoMod*, and most of the *AutoMod* keyboard commands work in *AutoView*.

4 AUTOVIEW ANIMATION CAPABILITIES

AutoView animation does not require a script. A model can be viewed and manipulated manually as the animation continues. To create an animation of a facility that automatically pans, zooms, and flows smoothly from scene to scene, however, we must create a script file. The description of the animation script is made in the camera description file (camera.des). There are two ways to describe a desired view: we can call a series of saved views or use *AutoView* syntax to describe moment-by-moment view changes. Time stamps (specific model times) and other control statements are used to create transitions and special effects.

There are three elements in a camera description file:

a. Views.
b. Motion syntax.
c. Control statements.

4.1 Views

A view is a static representation or picture of the model in some orientation in space. *AutoView* allows us to view the model at any angle and translation. *AutoView* views are created and saved through the following steps:

a. Use the View Control window or keyboard commands to change the view.

b. Save the view by typing a lowercase "l", followed by the name of the view.

It is important that we keep a record of the view names and a brief description of what each view contains for our own use, because we must supply the exact name of the view when writing a script file. If, however, we do not record a description of the views, we may look in the "modelname.vdf" file for a list of the views saved.

Creating and calling saved views is the simplest way to define animation in a script file. It is possible to create an interesting animation consisting entirely of views created and saved through View Control. In most cases we can use a saved view for focusing on an activity. To switch the focus to another area of the facility, we "pan" to a second saved view. We can also pan using *AutoView* control statements, but for simple view changes, saving two views and panning between them is the easiest approach.

4.2 Motion Syntax

Views can also be defined using *AutoView* syntax statements, but writing syntax requires much more effort than saving views. Some effects, however, such as following or riding specific loads or vehicles, can be achieved only by using *AutoView* syntax.

All *AutoView* syntax statements described in the script file contain "camera states." Camera states consist of the entity being viewed and a description of the camera position, velocity, acceleration and orientation. The camera position puts the camera in the world, the orientation points the camera, and the focal length determines the size of the entities in view. The camera state can be expressed either in relation to a path or entity or in absolute terms (relative to the "world"). The camera state can also contain a focal length that is similar to scale. Focal length, expressed relative to a position or relative to an entity, determines how large the entities appear in the view. The velocity and acceleration needed to move from one camera state to another can either be provided explicitly by the user or can be calculated by *AutoView* when the user indicates the two camera states and the duration of the move.

Making the camera follow a load or vehicle around a facility in the model is relatively easy. We simply specify the exact load or vehicle and the position of the camera relative to that load or vehicle. The view command indicates the position of the camera, and the camera command instructs the camera what to look at. For example, the following syntax might appear in the script file:

```
view focus relative to temp 5
camera relative to object AGVS:AGVS(2) at
  TCF Link1 pos 0, 0, 5
```

In this example, the view command focuses the camera on the origin location of entity temp 5. Temp 5 is the fifth load generated by *AutoMod*. The camera is positioned five units above the TCF, the terminal control frame Link1 of AGVS(2). The TCF is the location on the vehicle where the load resides. These two statements together indicate where the camera is located and what it is looking at. Now as the vehicle AGVS(2) and/or the load temp 5 move, the camera adjusts itself automatically.

We must always specify the load or vehicle on which we want the camera to focus. The camera cannot focus on all loads at once, nor can it choose one on which to focus. Watching the *AutoMod* model run will help us determine the load or vehicle we want to watch. After determining which activity we want to follow, we record the specific load or vehicle number using the standard load and vehicle status windows.

4.3 Control Statements

Control statements constitute the language of the *AutoView* camera description file. These statements represent the body of the camera description file and can be repeated as many times as desired. The following is a summary of each of the control statements:

grid	Toggles the grid on and off.
jump to	Jumps forward and back in time.
orthogonal	Changes view of animation to orthogonal.
pan to next view	Causes the camera to pan or sweep to next view.
pause	Causes the animation to pause.
perspective	Changes the view of animation to perspective.
repeat	Returns to the beginning of script.
solid	Changes all animation graphics to solid mode.
time scale	Changes the apparent speed of the animation.
turn	Turns the system <SYSTEM ID> picture <PICTURE ID> on or off.
view	Specifies the camera view.
wait for	Causes the animation to wait for the specified amount of time.
wire	Changes all animation graphics to wire frame mode.
window size full screen	Changes size of the simulation window to fit the screen.

5 CAMERA EXAMPLE

The following is a listing from a typical *AutoView* camera description file.

```
/*Indicates this is a camera description
file*/  camera description
/*Indicates distance unit of measure*/
distance units Feet
/*Indicates time unit of measure*/
time units Seconds
/*Indicates angle unit of measure*/
angle units degree

/*Makes window full screen*/
window size full screen
/*Current time scale * real time*/
time scale 1.00
/*Make animation graphics solid*/
solid
/*Make animation perspective*/
perspective

/*Turn off unwanted system objects*/
turn system agvs picture Text off
turn system agvs picture Paths off
turn system agvs picture Points off
turn system agvs picture Transfers off
turn system agvs picture Direction off
turn system proc picture Text off
turn system proc picture Blocks off

/*Jump animation clock to 400 time units*/
jump to time 400
/*Current view is 'top'*/
view top
/*Use view 'top' for 10 time units*/
time 410
/*Indicate pan to next specified view*/
pan to next view
/*Panning time is 10 time units*/
time 420
/*Next view is 'front'*/
view front
/*Hold view front for 10 time units*/
time 430
/*Time scale now 2 times real time*/
time scale 2
/*Jump animation clock to 830 time units*/
jump to time 830
/*Camera is at load 3, 5 distance units in
the z axis*/
view camera relative to object temp 3 pos
0,0,5
/*Camera focus is on agv 12, at origin of TCF
loadhere*/
view focus relative to object agvs:defagv(12)
at tcf loadhere pos 0,0,0
/*Stay focused on vehicle for 100 time
units*/
time 845
/*Repeat script file*/
repeat
```

6 SUMMARY

Simulation without the proper communication tools is pointless. The ability to communicate simulation results is as critical to a simulation project as the data used in the simulation model. Historically, it took an extremely long time to construct animation of a simulation model. New software packages such as *AutoView* have automated the process of constructing animated films, thus improving communication.

AutoView is a state-of-the-art animation tool that provides simulation analysts with a wide range of communication services. It allows analysts to construct walk-through animation of a factory or other operation in just hours, thus allowing managers to see their operations come to life and making it easier for them to understand the benefits and savings that can be obtained from their investments. Animation also helps analysts verify that model logic and code is functioning properly. Visualization helps convince non-proponents of simulation that a model is providing an accurate representation of their operations. Construct these animations provides both management and engineers with a visualization tool that is critical to project success.

REFERENCES

AutoSimulations, Inc. 1989. *AutoMod User's Manual.* Bountiful, UT.

AUTHOR BIOGRAPHY

RICHARD STAFFORD, Senior Simulation Analyst at AutoSimulations, Inc., has been involved in the development of over 200 simulation models in all facets of manufacturing. His activities at AutoSimulations include project management, model development, software development, and future product direction. Richard earned a B.S. in Industrial Engineering from the University of Utah.