

9-1998

# The VR Factory: Discrete Event Simulation Implemented in a Virtual Environment

Jason J. Kelsick  
*Iowa State University*

Judy M. Vance  
*Iowa State University, [jmvance@iastate.edu](mailto:jmvance@iastate.edu)*

Follow this and additional works at: [http://lib.dr.iastate.edu/me\\_conf](http://lib.dr.iastate.edu/me_conf)



Part of the [Computer-Aided Engineering and Design Commons](#)

---

## Recommended Citation

Kelsick, Jason J. and Vance, Judy M., "The VR Factory: Discrete Event Simulation Implemented in a Virtual Environment" (1998). *Mechanical Engineering Conference Presentations, Papers, and Proceedings*. 36.  
[http://lib.dr.iastate.edu/me\\_conf/36](http://lib.dr.iastate.edu/me_conf/36)

This Conference Proceeding is brought to you for free and open access by the Mechanical Engineering at Iowa State University Digital Repository. It has been accepted for inclusion in Mechanical Engineering Conference Presentations, Papers, and Proceedings by an authorized administrator of Iowa State University Digital Repository. For more information, please contact [digirep@iastate.edu](mailto:digirep@iastate.edu).

---

# The VR Factory: Discrete Event Simulation Implemented in a Virtual Environment

## **Abstract**

Virtual reality (VR) refers to an immersive, interactive, multi-sensory, viewer-centered, three-dimensional (3D) computer generated environment and the combination of technologies required to build such an environment (Cruz-Neira, 1993). Related to problems of engineering design and manufacturing, this new technology offers engineers the ability to work with computer models in a three-dimensional, immersive environment. This paper describes a virtual reality application where the results of a discrete event simulation of a manufacturing cell are integrated with a virtual model of the cell to produce a virtual environment. The program described in this paper, the VR Factory, allows the user to investigate how various changes to the manufacturing cell affect part production. This investigation is performed while immersed in a computer generated three-dimensional representation of the cell. This paper describes the creation of the VR model of the manufacturing cell, the animation of the environment and the implementation of the results of the discrete event simulation.

## **Keywords**

VRAC

## **Disciplines**

Computer-Aided Engineering and Design

## THE VR FACTORY: DISCRETE EVENT SIMULATION IMPLEMENTED IN A VIRTUAL ENVIRONMENT

Jason J. Kelsick

Judy M. Vance

Department of Mechanical Engineering  
Iowa Center for Emerging Manufacturing Technology  
Iowa State University  
Ames, Iowa 50011  
jkelsick@icemt.iastate.edu  
jmvance@iastate.edu

### ABSTRACT

Virtual reality (VR) refers to an immersive, interactive, multi-sensory, viewer-centered, three-dimensional (3D) computer generated environment and the combination of technologies required to build such an environment (Cruz-Neira, 1993). Related to problems of engineering design and manufacturing, this new technology offers engineers the ability to work with computer models in a three-dimensional, immersive environment. This paper describes a virtual reality application where the results of a discrete event simulation of a manufacturing cell are integrated with a virtual model of the cell to produce a virtual environment. The program described in this paper, the VR Factory, allows the user to investigate how various changes to the manufacturing cell affect part production. This investigation is performed while immersed in a computer generated three-dimensional representation of the cell. This paper describes the creation of the VR model of the manufacturing cell, the animation of the environment and the implementation of the results of the discrete event simulation.

### INTRODUCTION

Defined by Pritsker, "in its broadest sense, computer simulation is the process of designing a mathematical-logical model of a real system and experimenting with this model on a computer" (Pritsker, 1997). In discrete event simulation, the dependant variables must change at distinct times, thus forming events. In other words, the state of the simulated system can only change at event times. In use with manufacturing, discrete event simulations model part flow through a manufacturing process. The part flow is divided into a series of events with event times. The simulation can determine bottlenecks, machine tool usage, material handling problems, etc.

before they occur.

Computer simulation was realized as a potentially useful tool for industry in the late 1950's and early 1960's. It allowed industries to test configurations of manufacturing systems before purchasing and implementing the actual equipment. Since then, many simulation programming languages (SPLs) have been created and improved. In 1961 GASP (General Activity Simulation Program) was developed by Philip J. Kiviat at the Applied Research Laboratory (Nance, 1995). Two major SPLs used today are descendants of GASP. They are SLAM II (Simulation Language for Alternative Modeling), produced by Pritsker and Associates, Inc. (Pritsker, 1997), and SIMAN (SIMulation ANalysis), developed by C. Dennis Pegden. Both SPLs were developed in the late 1980's and have since then become major components of analysis and research for many industries.

The past few years have seen computer simulation development take a new direction. Previous versions of simulation software produced only text-based output. Today, visualization of the simulation is now possible because of the increased graphics capabilities of computers today. Pritsker and Associates, Inc. have created AweSim, an interface tool that creates a graphical user interface for SLAM II and also allows for the integration of outside programs and databases. Included in this version is the capability to create two-dimensional graphical animations. Animation helps the user visualize the simulation although it is limited to a two dimensional display. Another computer simulation package, developed by Deneb Robotics, is called QUEST® (QUEuing Event Simulation Tool). QUEST®, includes a three-dimensional graphical animation that allows for a more immersive environment than the two-dimensional animation provided by AweSim.

The VR Factory, the program described in this paper, is an animated, three-dimensional model of a manufacturing work

cell. A manufacturing process is simulated in the VR Factory by implementing a discrete event simulation of the process. A simulation model of the work cell, created using SLAM II, provides the discrete event simulation data file that the virtual environment utilizes. The topic of this paper is how the VR Factory was created and how the challenges in the creation were addressed. The steps required in the creation of the VR Factory include the modeling and animation of the machines, tools, parts, etc., and the implementation of the results of the discrete event simulation.

## THE VR FACTORY ENVIRONMENT

### Peripherals

The VR Factory can be viewed in several different environments: in a head-mounted display (HMD), on a stereoscopic projection screen, or in Iowa State University's C2, comprised of four stereoscopic walls and a three-dimensional sound system. When viewed on the projection screen or in the C2, stereo vision is obtained through CrystalEyes stereo shutter glasses. Interaction with the virtual environment (created by any of the display devices mentioned above) is achieved through a Fakespace PINCH™ Glove. The Fakespace PINCH™ Glove records contact between a user's so various hand gestures can be used to control movement and virtual menu selections. The user's viewpoint and hand position are tracked with Ascension Flock of Birds™ magnetic trackers. When VR peripherals are not available, the user can interact with the program using a standard monitor and mouse.

### Navigation and Interaction

Since the factory floor space is larger than the workspace of the virtual environment, tracking the viewpoint position with the Flock of Birds™ alone does not allow the user to examine the entire work cell. Extra navigation is needed to move to a viewpoint outside the workspace of the virtual environment. The PINCH™ Glove gestures not only enable this extra navigation; they also allow navigation to be independent of head tracking. The user can look off to one side, yet use the PINCH™ Glove to navigate forward.

To navigate, the user reaches out, touching the index finger to the thumb, and pulling inward while holding the finger and thumb together. A good analogy would be grabbing a rope and pulling yourself along (See Figure 1). Other aspects of the navigation are touching the middle finger to the thumb rotates the user's viewpoint clockwise and touching the ring finger to the thumb rotates the user's viewpoint counter-clockwise.

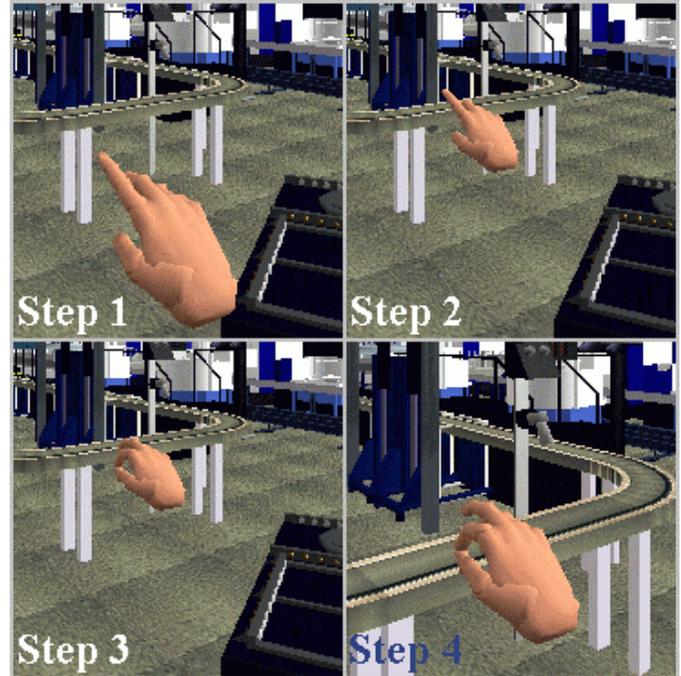


Figure 1. Navigation in the Virtual Environment

Interaction with the VR Factory is through a three-dimensional menu that can be positioned anywhere in the virtual space. To make the menu appear the user makes a fist (making contact with the tips of all four fingers to the palm of the hand). The options on the menu list different possible simulations of the same factory work cell (See Figure 2). To choose a particular simulation, the user must intersect the virtual hand with the menu option and make a gesture with the index finger and thumb. Once the option has been chosen, the menu disappears and the simulation begins.



Figure 2. Virtual Menu

Another aspect of the interaction with the virtual environment is the identification of each part and its characteristics. The user can open an identification table listing the part's type, its current station, the time left at the station, and the next station the part will go to by touching the pinky finger to the thumb. This table will stay with the user while the user navigates through the VR Factory. By intersecting the virtual hand with a part in question, the table is updated to show the current statistics of the part.

Probably the most important and obvious aspect of the visualization of a simulation is the creation of the geometric models (three-dimensional representations) of the machines, tools, parts, etc., involved in the simulation. The following section describes the process in which the geometric models for the VR Factory were created.

### Creation of the Factory Models

The first step in creating the geometric models was acquiring all of the necessary measurements and dimensions. Floor plans and dimensional drawings of the machines were used as references, but since the VR Factory is modeled after a factory located in the area, dimensions were also acquired by measuring the actual objects.

CAD software (Pro/Engineer®) and modeling software (World Up™ Modeler, MultiGen® II) were used to build the models. In these modelers, material properties such as color, shininess, and emissive properties were assigned to the models. By manipulating these properties, a model becomes more realistic. For example, if a particular machine is made of metal, the shininess property of the representative model is increased (See Figure 3).

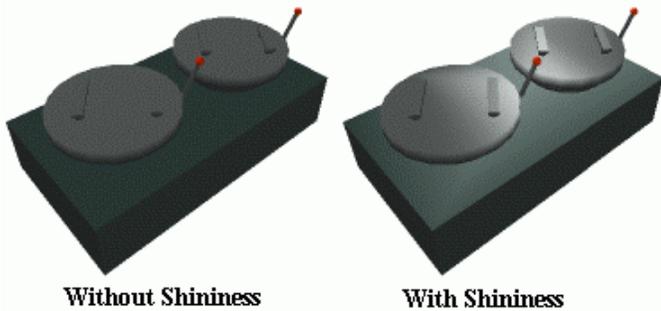


Figure 3. Material Properties Example

Textures were also applied to the models in the modelers. The textures were created in two main ways. They were either gathered by taking pictures of the actual machines and tools or created with Adobe PhotoShop. Textures, like material properties, add realism to the virtual environment. In fact, they can add much more realism to the models than the material properties. Figure 4 shows how a few textures can add realism to a

model of a computer monitor and its surroundings. However, textures require larger amounts of memory, unlike material properties, and therefore slow down the display of the computer. If too many textures are applied to the models, the program will slow down and not run in real time. This problem, realism vs. the speed of the program, is not only apparent in this example but also in other aspects of the creation of the geometric models.

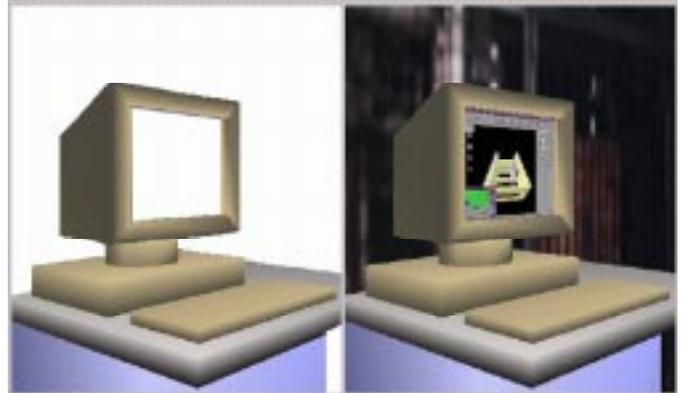


Figure 4. Texture example

One such aspect is the number of polygons that make up a model. Figure 5 shows two machining center doors. Both doors have the same dimensions yet the door on the left has hundreds more vertices. Since a computer stores the position and orientation of every vertex, using the door on the left instead of the right would be a dramatic increase in the amount of needed memory. Thus, creating a model with the smallest amount of polygons needed to keep the model realistic was a constant consideration. A common solution to reducing the number of polygons in the model was to decide if a certain aspect of the model would be noticed or seen at all in the simulation. If it would not be noticed it was deleted from the model.

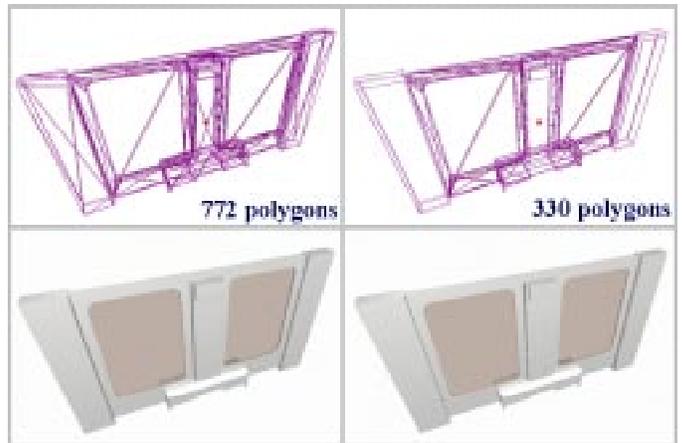


Figure 5. Number of Polygons Example

Once the models were created, they were loaded into the program and translated to either the position defined by the floor plan or, if that was not possible, in relation to other objects in the virtual factory. The software used to manage the virtual environment was the C/C++ toolkit called WorldToolKit™ from Sense8. As more and more models were loaded into the program it was becoming apparent that the models were still too complex (too many polygons). This in turn caused the program to slow down. To maintain speed, level of detail (LOD) models were introduced. LOD is "modeling the same object at different detail levels and the appropriate one is chosen for display based on some viewing criteria and system performance" (Chen, 1995). In other words, if the viewpoint of the user is close to a particular object in the factory, a detailed version of the object is displayed. If the user is far enough away from the object so that the smaller features could not be distinguished, a simpler version of the model is displayed (Fleischer, 1995). With the combination of reduction of polygons, limited use of textures, and LOD, a more efficient model of the factory work cell was created.

### **Animation of the Factory Models**

Once the geometric models in the VR Factory were constructed and placed in the virtual environment, the implementation of the simulation could be performed. Animation of the objects in the virtual environment required knowledge of how the actual machines and tools worked. Most of this knowledge came from observing the real factory work cell.

The program's structure was based chiefly on how each individual object in the virtual world would be animated. Each geometric model in the VR Factory is considered an entity in the program known as a node and the structure of these nodes is called the node hierarchy. The node hierarchy's structure is similar to the structure of a family tree with each node having a parent node and possibly a child node. The child node would then inherit the motion of the parent node.

With the knowledge of how the machines and tools in the work cell worked, the models of these machines and tools were structured for proper animation. For example, a door on a machining center was loaded into the program as a child node of the machining center node. This allowed for the door to be animated independently of the machining center, but if the location of the machining center were moved, the door would also move.

The models of the parts being manufactured in the VR Factory were initially not part of the node hierarchy. As the simulation runs, the parts are attached (become child nodes) and detached from other nodes. When the part is sitting on a particular pallet it is attached to the pallet. If the pallet were to move, the part would also move. Detaching and then attaching the part to another node would then make the part move with this node. In this manner, the parts are "carried" throughout the manufacturing process.

Collision of the geometric models was also a consideration in the animation of the VR Factory. The program, not designed with collision detection, allows for geometric objects to occupy the same space. This collision detection had to be done visually and corrected manually because using the computer to analyze collision detection would reduce the speed of the program.

Time was a very important factor in the animation of the VR Factory. It is the major independent variable in the discrete event simulation of the manufacturing process. Therefore, it is the controlling variable in the VR Factory. An example of this would be when a part is moved from one location to another location. At a specific time, defined by the results of the simulation, the part is translated to the new location. This means the program continually checks the time and when the clock arrives at the time given by the simulation, the part is translated. In other words, animation is activated and deactivated at specific points in time. By setting up the animation in this manner, the implementation of the discrete event simulation results is fluid.

### **Implementation of the Results of the Discrete Event Simulation**

In order to implement the results of a discrete event simulation into a virtual environment, the output capabilities of the simulation software (SLAM II) and the input capabilities of the software enabling a virtual environment (WorldToolKit™) must be clearly defined. In the case of the VR Factory, a text based data file was exported from SLAM II and easily imported into the WorldToolKit™ virtual environment. The structure of such a file was not crucial because the VR Factory could be adjusted to import any information. Thus, the main influence on its structure was the exporting capabilities of the simulation software. With that into consideration, the most important questions faced in this implementation were: What input variables are needed in the virtual environment and what output variables can be supplied by the simulation software?

As mentioned earlier, time is the major independent variable in the simulation of the manufacturing process. This means at specific times the state of the part changes. For example, a part may be transported to a machining center, machined, transported to an inspection station, inspected, and then returned to storage. Each part of the example above is defined by a specific starting time. If a virtual environment were created to visualize this process, the necessary information about the part's characteristics and how it moves through the process should already be defined in the VR Factory. The time at which the part changes its state and the destination of the part, call them attributes, are determined from the simulation. This is the information that is loaded into the VR Factory from the results of the discrete event simulation created using SLAM II.

In this particular case, the results of the discrete event simulation were obtained by placing markers throughout the

simulation's process at the points where each part changes its state. When the simulation (SLAM II) was executed, these markers identified when to record the elapsed time and/or the part's destination into a data file. The data file thus became a large matrix in which each row identified a particular part and its attributes. This data file is then incorporated into the file system of the VR Factory for the VR Factory to use when it is executed.

### Executing the VR Factory

An initial simulation is created using SLAM II and the results are stored and upon execution of the VR Factory, they are read into the virtual environment. When the program begins, the user is placed inside the synthetic environment composed of the factory floor. By using the navigation mentioned in the section Navigation and Interaction, the user is able to move about the VR Factory. This navigation, along with the tracking of the user's viewpoint, allows the user to get close to the models in the factory and view them from any angle, including from beneath the models. To start a manufacturing simulation, the user selects a particular simulation from the virtual menu and the manufacturing process begins. At this point the user can not only inspect the machines and tools in operation, but also follow any part through a complete process. When following the part through its process, the user can identify its characteristics by viewing the virtual table. The user can at any time begin a new simulation by making a selection from the virtual menu and the process is started over.

### CONCLUSIONS

Integrating results from a discrete event simulation into a virtual factory model provides a three-dimensional environment in which to examine these results. The VR Factory allows a user to be completely immersed in a functioning factory work cell. Through this visualization tool, the user might see where a problem could arise instead of tracing a problem through charts and graphs of the simulation. Because of this, the VR Factory was successful representation of a visualization tool for discrete event simulations.

### FUTURE WORK

There are many aspects of the visualization of a simulation that the VR Factory has not fully explored and are currently being developed. One is implementing different scenarios developed through the use of SLAM II. Examples of the scenarios would include having a different number of machining centers, other machines, or operators of the machines. The user would then be allowed to interactively select the scenario of choice while in the virtual environment.

Another feature planned for the VR Factory is allowing the user to query certain aspects of the environment. Displaying a virtual "tablet" when the user touches a machine, tool, part,

etc. could do this. The "tablet" would list pertinent information about the object that would be useful to the user.

Once these additional aspects are added to the VR Factory, a study will be performed to determine the benefits of VR in the visualization of manufacturing simulations. The VR Factory could be compared against traditional workstation-based simulation. Results of this type of study could justify the use of VR as a visualization tool for simulations.

### ACKNOWLEDGEMENTS

This work is supported through funding of the National Science Foundation project DMI-9525998. Equipment for this project is supplied by the Iowa Center for Emerging Manufacturing Technology. The authors would also like to thank Professor Cheryl Moller-Wong of Iowa State University for her guidance and Lori Melaas for the SLAM II and AweSim results.

### REFERENCES

- Adobe PhotoShop User's Guide*, version 4.0, Adobe Systems, Inc., 1996.
- Angster, S., Gowda, S., and Jayaram, S., Using VR for Design and Manufacturing Applications: A Feasibility Study, *Proceedings of the 1996 ASME Design Engineering Technical Conferences*, Irvine, CA, CIE-A-W4, August, 1996.
- Banks, J., Carson, J., and Nelson, B., *Discrete-Event Simulation*, Prentice Hall, Upper Saddle River, NJ, 1996.
- Barfield, W. and Furness, T., *Virtual Environments and Advanced Interface Design*, Oxford University Press, New York, NY, 1995.
- Brown, R., An Overview of Virtual Manufacturing Technology, *Proceedings of the 1997 ASME Design Engineering Technical Conferences*, Sacramento, CA, DETC97/DFM-4362, September, 1997.
- Chen, S. E., QuikTime® VR-An Image\_Based Approach to Virtual Environment Navigation, *Computer Graphics: Proceedings of SIGGRAPH '95*, Los Angeles, CA, August 1995.
- Cruz-Neira, C., Virtual Reality Overview, *ACM SIGGRAPH '93 Notes: Applied Virtual Reality*, ACM SIGGRAPH '93 Conference, Anaheim, California, August 1-6, 1993.
- Fleischer, K., Laidlow, D., Currin, B., and Barr, A., Cellular Texture Generation, *Computer Graphics: Proceedings of SIGGRAPH '95*, Los Angeles, CA, August, 1995.
- Foley, J., van Dam, A., Feiner, S., and Hughes, J., *Computer Graphics: Principles and Practice*, Addison-Wesley Publishing Company, Reading MA, 1995.
- Gupta, R., Survey on Use of Virtual Environments in Design and Manufacturing, *Proceedings of the 1996 ASME Design Engineering Technical Conferences*, Irvine, CA, CIE-A-W4, August, 1996.
- MultiGen® II User's Guide*, version 1.2, San Jose, CA, September 1996.

Nance, R. E., Simulation Programming Languages: An Abridged History, *Proceedings of the 1995 Winter Simulation Conference*.

Pritsker, A. B., O'Reilly, J., and LaVal, D., *Simulation With Visual SLAM and AweSim*, Systems Publishing Corporation, West Lafayette, Indiana, 1997.

*Pro/Engineer® Drawing User's Guide*, Waltham, MA, 1997.

QUEST simulation software, Deneb Robotics, Inc.

Sense8 Corporation, *World Up™ User's Guide*, Mill Valley, CA, May, 1997.

Sense8 Corporation, *WorldToolKit™ Reference Manual Release 6*, Mill Valley, CA, 1996.

Varshney, A., El-Sana, J., Evans, F., Darsa, L., Costa, B., and Skiena, S., Enabling Virtual Reality for Large-Scale Mechanical CAD Datasets, *Proceedings of the 1997 ASME Design Engineering Technical Conferences*, Sacramento, CA, DETC97/DFM-4371, September, 1997.